

# Journal of Astronomical Telescopes, Instruments, and Systems

AstronomicalTelescopes.SPIEDigitalLibrary.org

## Data reduction pipeline for the CHARIS integral-field spectrograph I: detector readout calibration and data cube extraction

Timothy D. Brandt  
Maxime Rizzo  
Tyler Groff  
Jeffrey Chilcote  
Johnny P. Greco  
N. Jeremy Kasdin  
Mary Anne Limbach  
Michael Galvin  
Craig Loomis  
Gillian Knapp  
Michael W. McElwain  
Nemanja Jovanovic  
Thayne Currie  
Kyle Mede  
Motohide Tamura  
Naruhisa Takato  
Masahiko Hayashi

Timothy D. Brandt, Maxime Rizzo, Tyler Groff, Jeffrey Chilcote, Johnny P. Greco, N. Jeremy Kasdin, Mary Anne Limbach, Michael Galvin, Craig Loomis, Gillian Knapp, Michael W. McElwain, Nemanja Jovanovic, Thayne Currie, Kyle Mede, Motohide Tamura, Naruhisa Takato, Masahiko Hayashi, "Data reduction pipeline for the CHARIS integral-field spectrograph I: detector readout calibration and data cube extraction," *J. Astron. Telesc. Instrum. Syst.* **3**(4), 048002 (2017), doi: 10.1117/1.JATIS.3.4.048002.

**SPIE.**

# Data reduction pipeline for the CHARIS integral-field spectrograph I: detector readout calibration and data cube extraction

Timothy D. Brandt,<sup>a,\*</sup> Maxime Rizzo,<sup>b</sup> Tyler Groff,<sup>b</sup> Jeffrey Chilcote,<sup>c</sup> Johnny P. Greco,<sup>d</sup> N. Jeremy Kasdin,<sup>e</sup> Mary Anne Limbach,<sup>e</sup> Michael Galvin,<sup>e</sup> Craig Loomis,<sup>d</sup> Gillian Knapp,<sup>d</sup> Michael W. McElwain,<sup>b</sup> Nemanja Jovanovic,<sup>f</sup> Thayne Currie,<sup>g</sup> Kyle Mede,<sup>h</sup> Motohide Tamura,<sup>i,j,k</sup> Naruhisa Takato,<sup>g</sup> and Masahiko Hayashi<sup>i</sup>

<sup>a</sup>University of California, Santa Barbara, Department of Physics, Santa Barbara, California, United States

<sup>b</sup>NASA Goddard Space Flight Center, Greenbelt, Maryland, United States

<sup>c</sup>Stanford University, Department of Physics, Palo Alto, California, United States

<sup>d</sup>Princeton University, Department of Astrophysical Sciences, Princeton, New Jersey, United States

<sup>e</sup>Princeton University, Department of Mechanical and Aerospace Engineering, Princeton, New Jersey, United States

<sup>f</sup>California Institute of Technology, Department of Astronomy, Pasadena, California, United States

<sup>g</sup>National Astronomical Observatory of Japan, Subaru Telescope, Hilo, Hawaii, United States

<sup>h</sup>Rakuten, Inc., Setagaya, Tokyo, Japan

<sup>i</sup>University of Tokyo, Department of Astronomy, Hongo, Tokyo, Japan

<sup>j</sup>National Astronomical Observatory of Japan, Mitaka, Tokyo, Japan

<sup>k</sup>Astrobiology Center of NINS, Osawa, Mitaka, Tokyo, Japan

**Abstract.** We present the data reduction pipeline for CHARIS, a high-contrast integral-field spectrograph for the Subaru Telescope. The pipeline constructs a ramp from the raw reads using the measured nonlinear pixel response and reconstructs the data cube using one of three extraction algorithms: aperture photometry, optimal extraction, or  $\chi^2$  fitting. We measure and apply both a detector flatfield and a lenslet flatfield and reconstruct the wavelength- and position-dependent lenslet point-spread function (PSF) from images taken with a tunable laser. We use these measured PSFs to implement a  $\chi^2$ -based extraction of the data cube, with typical residuals of  $\sim 5\%$  due to imperfect models of the undersampled lenslet PSFs. The full two-dimensional residual of the  $\chi^2$  extraction allows us to model and remove correlated read noise, dramatically improving CHARIS's performance. The  $\chi^2$  extraction produces a data cube that has been deconvolved with the line-spread function and never performs any interpolations of either the data or the individual lenslet spectra. The extracted data cube also includes uncertainties for each spatial and spectral measurement. CHARIS's software is parallelized, written in Python and Cython, and freely available on github with a separate documentation page. Astrometric and spectrophotometric calibrations of the data cubes and PSF subtraction will be treated in a forthcoming paper. © The Authors.

Published by SPIE under a Creative Commons Attribution 3.0 Unported License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI. [DOI: [10.1117/1.JATIS.3.4.048002](https://doi.org/10.1117/1.JATIS.3.4.048002)]

Keywords: data processing; infrared spectroscopy; multispectral imaging; deconvolution; tunable lasers; point-spread functions.

Paper 17023 received May 26, 2017; accepted for publication Oct. 27, 2017; published online Nov. 24, 2017.

## 1 Introduction

Due to the advent of large format detectors, integral-field spectrographs (IFSs) have become an increasingly popular class of astronomical instrumentation. IFSs are hybrids of traditional imaging cameras and slit spectrographs; they obtain a spectrum from each spatial element in a two-dimensional (2-D) field-of-view for an  $(x, y, \lambda)$  data cube. The first realization of an IFS used a bundle of fibers to create a pseudoslit,<sup>1</sup> while TIGER<sup>2</sup> was the first IFS to use a lenslet array. Modern IFSs generally use either fiber bundles (MaNGA)<sup>3</sup> or image slicers (SINFONI, MUSE, and NIRSpec)<sup>4–6</sup> to rearrange the field-of-view into a long pseudoslit or lenslet arrays (OSIRIS)<sup>7</sup> to reimage spatial elements into small spots suitable for dispersion.

IFSs have become especially popular tools for high-contrast imaging. Diffraction speckles have a different chromatic behavior from astrophysical sources; an IFS data cube can exploit this

to achieve higher contrasts.<sup>8,9</sup> An IFS also naturally enables the extraction of a planet or brown dwarf's spectrum, providing a probe of the object's temperature, chemistry, and gravity.<sup>10–14</sup> IFSs combined with second-generation adaptive optics systems are now operational on Gemini South (GPI),<sup>15</sup> the VLT (SPHERE),<sup>16</sup> and Palomar (Project 1640).<sup>17</sup> These new high-contrast instruments have recently discovered and characterized the low-mass companion to 51 Eri.<sup>18</sup> Future NASA mission studies such as Exo-C<sup>19</sup> and the WFIRST Coronagraph Instrument have baselined high-contrast IFSs as their science cameras.<sup>20,21</sup>

Data reduction and processing for IFSs have long presented problems. The reduction pipeline for GPI<sup>22,23</sup> is an ongoing, years-long effort partially built on legacy software from OSIRIS (whose pipeline also remains, to some degree, a work in progress). This is the result of many complexities inherent in IFS data. There are now two flatfields (one for the detector and one for the illumination of the fibers, lenslets, or sliced image plane). For a lenslet-based IFS, the point-spread functions

\*Address all correspondence to: Timothy D. Brandt, E-mail: [tbrandt@ucsb.edu](mailto:tbrandt@ucsb.edu)

(PSFs) of the input optics and the lenslets are both important. The finite size of the lenslet PSFs means that neighboring spectra partially overlap one another; this must be corrected or accounted for during the extraction. An IFS requires the wavelength and spectrophotometric calibrations of a spectrograph as well as the astrometric calibration of an imager.

This paper presents the data reduction pipeline for the CHARIS IFS on the Subaru telescope. CHARIS, the Coronagraphic High Angular Resolution Imaging Spectrograph, is a lenslet-based near-infrared IFS located on the Nasmyth platform behind the adaptive optics systems AO188<sup>24</sup> and SCExAO.<sup>25</sup> Section 2 summarizes the design and properties of the IFS, while the rest of the paper presents the software that extracts the data cube. Section 3 discusses the construction of the pixel-by-pixel count rates from a sequence of raw reads, Sec. 4 discusses our calibration procedure and associated data products, and Sec. 5 presents our algorithms for extracting the data cube. Section 6 summarizes the software's parameters and settings, and Sec. 7 shows the software's performance. We discuss and conclude with Sec. 8.

This software package constructs the data cube and its inverse variance from a sequence of CHARIS reads. We defer the necessary steps of image registration and spectrophotometric and astrometric calibration to a separate software package that is currently under development. These steps depend on the SCExAO system in front of CHARIS and the calibrations changed when SCExAO was rebuilt in July of 2016; they also rely on a system of induced satellite spots that still must be manually controlled by the SCExAO team.<sup>26</sup> These elements of the software, in addition to algorithms for angular and spectral differential imaging, will operate only on the data cubes produced by this pipeline and will be presented in a follow-up paper.

## 2 CHARIS Integral-Field Spectrograph

CHARIS is a new high-contrast IFS for the Subaru Telescope. Its scientific, conceptual, optical, and mechanical designs are summarized by McElwain et al.,<sup>27</sup> Peters et al.,<sup>28</sup> Galvin et al.,<sup>29</sup> and Peters-Limbach et al.,<sup>30</sup> respectively. Groff et al.<sup>31</sup>

summarized laboratory testing performed after CHARIS was built but before it was transported to the summit. This section briefly reviews the basic parameters and observing modes of CHARIS; we refer the reader to these other papers and to a forthcoming instrument paper for details.

CHARIS is a lenslet-based diffraction-limited spectrograph operating in the near-infrared. Table 1 summarizes its basic as-built properties. CHARIS uses one of two prisms behind the lenslet array to disperse the light from each lenslet into a  $\sim 30$ -pixel-long microspectrum. The detector image consists of about  $135 \times 135$  of these microspectra, each containing the light incident on a single 16.4 mas square lenslet, for a  $\sim 2''.2 \times 2''.2$  field-of-view.

CHARIS offers five observing modes, three with a high-resolution prism and two with a low-resolution prism. We measure its as-built spectral resolution  $R$  using the definition

$$R = \lambda / \delta\lambda = \left( 2 \frac{d \ln \lambda}{dx} \right)^{-1}, \quad (1)$$

with  $x$  in pixels, so the dispersion is a wavelength shift per two pixels [slightly larger than the full width at half maximum (FWHM) of the lenslet PSFs]. With this definition,  $R$  with the high-resolution prism varies from  $\sim 85$  at the short end of  $J$  to  $\sim 65$  in the middle of  $H$  to  $\sim 85$  at the long end of  $K$ . The low-resolution prism has an  $R$  that varies from about 18 to 22 across the  $J$ ,  $H$ , and  $K$  bands. The high-resolution mode uses either a  $J$ ,  $H$ , or  $K$  band filter, while the low-resolution prism may be used either with a broadband or a neutral density (ND) filter. The broadband filter has nearly unit transmission from 1.15 to 2.38  $\mu\text{m}$  and sharp cutoffs toward both shorter and longer wavelengths. The ND filter is ND3 ( $10^{-3}$  transmission) between 1.15 and 2.38  $\mu\text{m}$  and opaque at other wavelengths; it is inside the CHARIS dewar to reduce the thermal  $K$ -band background. The ND filter is intended to allow observers to obtain unsaturated images of bright stars. CHARIS's location at Subaru's Nasmyth platform gives it better stability than GPI or Project 1640, while its extremely flat dispersion across the  $J$ ,  $H$ , and  $K$  bands is unique among high-contrast IFSs. CHARIS's use of pinholes on the lenslet array and its relatively generous spacing of microspectra result in very low spectral cross-talk<sup>31</sup> and enable us to model and remove correlated read noise (Sec. 5.3).

CHARIS is controlled by a Linux-based software and is integrated into the software environment of the observatory. It has only three moving parts to be controlled during observations: a shutter, the five-slot filter wheel ( $J$ ,  $H$ ,  $K$ , broadband, and ND), and the three-position prism slider (low resolution, high resolution, and empty). The only other command is to reset and read out the detector. The rest of this paper presents the software for reconstructing the individual lenslet microspectra, i.e., the data cube, from these reads.

## 3 From the Reads to a Ramp

CHARIS's detector is a Hawaii2-RG (H2RG), a HgCdTe CMOS device in which each pixel has its own amplifier. The pixel is read out by measuring the voltage across it relative to a reference voltage in the system. In CHARIS's configuration, 32 readout channels each read pixels at a rate of 100 kHz. Including some down time as the readout proceeds to the next row of pixels, it takes 1.47 s to read out the full  $2048 \times 2048$  pixel array. Resetting the detector is done pixel-by-pixel

**Table 1** Basic CHARIS parameters.

Parameter	Value
Detector	2048 $\times$ 2048 H2RG
No. of lenslets	135 $\times$ 135
Lenslet size	16.4 mas
Field-of-view	2''.2 $\times$ 2''.2
Wavelength coverage	1.15 to 2.38 $\mu\text{m}$
Microspectrum length	$\sim 30$ pixels
$R = \lambda / \delta\lambda$ (2 pixels)	$\sim 20$ (low res), $\sim 75$ (high res)
Available modes	$J$ , $H$ , or $K$ at $R \sim 75$ $J + H + K$ or ND <sup>a</sup> at $R \sim 18$

<sup>a</sup>Filter is ND3 ( $10^{-3}$  transmission) from 1.15 to 2.4  $\mu\text{m}$ , opaque at other wavelengths.

and also takes 1.47 s. H2RG detectors have some generic shortcomings, including persistence after exposure to a bright source<sup>32</sup> and  $1/f$  read noise strongly correlated among readout channels.<sup>33</sup>

CHARIS always saves every one of its reads, and this sequence of raw reads is then saved to disk. The first step of the data extraction is to fit for the count rate at each pixel from the raw reads. In the limit of a linear pixel response and the dominance of read noise over photon noise, the best-fit count rate  $C_i$  in pixel  $i$  may be derived from a  $\chi^2$  fit to the sequence of reads  $j$

$$\chi_i^2 = \sum_{\text{reads } j=1}^N \frac{(j\delta t \cdot C_i + b_i - F_{ij})^2}{\sigma^2}, \quad (2)$$

where  $\sigma^2$  is the variance from the read noise,  $b_i$  is the pixel's reset value,  $j\delta t$  is the time from reset at which pixel  $i$  was read out for the  $j$ 'th time, and  $F_{ij}$  is the number of counts in read  $j$  of pixel  $i$ . Minimizing  $\chi^2$  with respect to  $C_i$  and  $b_i$ , we find that the best-fit count rate  $C_i$  is given by a linear combination of the reads  $j$ , and the readout is called up-the-ramp (UTR)<sup>34</sup>

$$C_i\delta t = \frac{12}{N^3 - N} \sum_{j=1}^N \left( j - \frac{N+1}{2} \right) F_{ij}, \quad (3)$$

where  $N$  is the total number of reads. UTR, or a variant using only some of the available reads, is now commonly used to read out infrared arrays both on the ground and in space.<sup>23,35–37</sup>

In the limit of uncorrelated read noise, UTR improves on the signal-to-noise ratio (SNR) of correlated double sampling (CDS), the normalized difference of the first and last reads, by a factor

$$\frac{\text{SNR}_{\text{UTR}}}{\text{SNR}_{\text{CDS}}} = \sqrt{\frac{N(N+1)}{6(N-1)}}. \quad (4)$$

If there are only two reads, i.e.,  $N = 2$ , UTR and CDS are equivalent, and the ratio in Eq. (4) is unity. It is never less than one and increases asymptotically as  $\sqrt{N/6}$  or  $\sqrt{t_{\text{exp}}/(9\text{s})}$  assuming 1.5 s/read. When photon noise dominates, UTR is asymptotically noisier than CDS by about 23%.<sup>38</sup> This may be corrected by dynamically choosing different weights for each read at each pixel (e.g., Ref. 39), but such an approach is not easily compatible with our nonlinear fit (Sec. 3.2). In the high signal-to-noise regime, we are limited by the fidelity of our models of the microspectra rather than by photon noise.

We compute a variance on a given pixel's count rate from the variance in the count rates of the reference pixels in its channel. We then add photon noise assuming our configured gain of  $2 \text{ e}^-/\text{count}$  to be correct. We have verified this gain using a frame-to-frame scatter in count rate as a crude measure of shot noise. In addition, we allow the user to enforce an additional error equal to a fixed fraction (with a suggested value of a few percent) of the count rate seen by each pixel. This additional error accounts for our imperfect model of the microspectra and produces reduced  $\chi^2$  values close to unity when fitting these models to the ramps; we add it in quadrature to the other errors. The variances are saved as inverse variances, so bad pixels may be masked by giving them an inverse variance of zero.

Cosmic ray hits are rare in short CHARIS exposures, and we do not implement cosmic ray rejection within our ramps.<sup>38,40</sup> Instead, we mask pixels in which a single read exceeds that pixel's mean count rate by a factor of 5 and its mean read noise by a factor of 10. Less than 0.03% of pixels are masked in this way in a typical 2-min exposure.

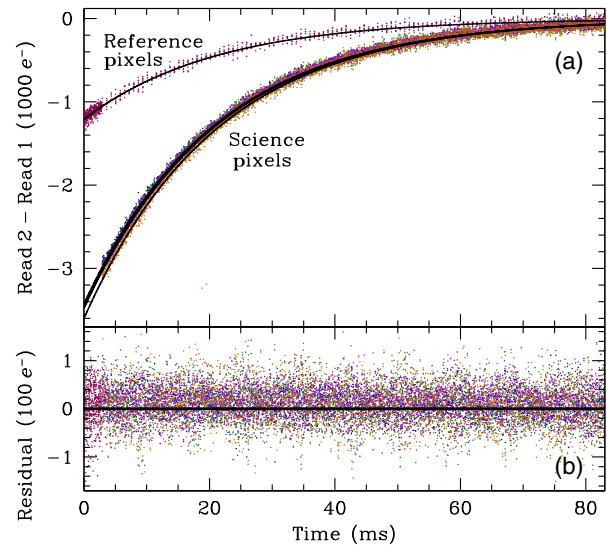
The CHARIS software generally uses the UTR coefficients in Eq. (3). The following subsections discuss the removal of an artifact in the first read of a CHARIS ramp and our handling of nonlinearity and saturation. We then briefly discuss the read noise properties of our ramps. In Sec. 5.3, we will take advantage of the relatively low dimensionality of much of the read noise, fitting it out when extracting data cubes.

### 3.1 Correcting Artifacts in the First Read

The first read in a CHARIS ramp is contaminated by an exponential decay of the reference voltage. Figure 1 shows the difference between the first and second reads of a dark frame, while Fig. 2(a) shows the lower-left corner of a 17-read ramp taken in CHARIS's low-resolution mode without correcting for this decay. Given CHARIS's baffling and low level of dark current, the mean count rate in a dark frame is much lower than the read noise; Fig. 1 shows a noisy decay to zero. For the time series in Fig. 1, the 2-D readout of each channel has been mapped back to one dimension assuming a pixel rate of 100 kHz and an 8- $\mu\text{s}$  downtime among reading rows of pixels on the detector. This exponential decay has a time constant of  $\sim 21 \text{ ms}$  and becomes negligible by 100 to 200 ms ( $\sim 10\%$  of the 1.47 s full-frame readout time). Similar artifacts have previously been noted in H2RGs and in related detectors.<sup>41,37</sup>

The black lines in Fig. 1 take the form

$$y = A_i \exp[-t/t_0], \quad (5)$$



**Fig. 1** (a) The exponential decay of the reference voltage in the first read, shown via the difference between the first and second reads in a dark frame. Four channels are shown in green, violet, blue, and orange, with the reference pixels in burgundy. The solid black lines are exponential fits with a shared decay constant of about 21 ms, and (b) shows the residuals. The residuals deviate slightly from zero due to low-frequency noise on the detector. Removing the exponential decay of the reference voltage allows us to use the first read in all CHARIS ramps.



where the time constant  $t_0 = 21.1$  ms is the same for all of the lines and the  $A_i$  are fitted separately to each readout channel  $i$  (and to the reference pixels). The lower panel shows the residuals, indicating a good fit with some remaining low-frequency read noise. Removing the first-read artifact requires fitting 34 parameters in all: one time decay constant  $t_0$ , 32 amplitudes for the 32 readout channels, and an additional amplitude for the reference pixels. We perform the fit as follows.

In a CHARIS ramp, the difference between the counts in the first and subsequent reads is the sum of the read noise, the exponential decay of the reference voltage, and the photon rates scaled by the gain. We first estimate the photon count rates by fitting a ramp only to the second and subsequent reads, ignoring the contaminated first read. We then use the fitted pixel-by-pixel count rates and reset values to obtain a modeled number of counts at the first read. The difference between the actual and modeled counts in the first read is the sum of read noise and the reference voltage artifact that we wish to remove. We must then perform a 34-parameter nonlinear fit. Luckily, the fit is only nonlinear in a single parameter, the decay constant  $t_0$ . Once the decay constant is fixed, the 34-parameter nonlinear optimization becomes 33 decoupled one-parameter linear optimizations. By solving these linear problems for each value of  $t_0$ , we may reduce the problem to a one-dimensional (1-D) nonlinear optimization for  $t_0$ . We first guess the value of  $t_0$  from the known behavior of the CHARIS detector and then iteratively fit parabolas to converge to the best-fit decay constant. This is equivalent to Newton–Raphson iteration on  $d\chi^2/dt_0$ . Once we have the decay constant, the other 33 parameters may be obtained by straightforward linear optimizations. The entire process for a typical CHARIS ramp takes a few hundred milliseconds on a laptop computer.

Figure 2(b) shows the 17-read ramp of (a) after removing the exponential decay of the reference voltage. There are no longer any artifacts visible, and the first read may now be used to increase the integration time on-source and reduce the read noise. The first read consists of  $2048 \times 2048$  pixels, many

orders of magnitude larger than the 34 parameters describing the reference voltage artifact; fitting out this artifact uses a negligible amount of the information available in the first read.

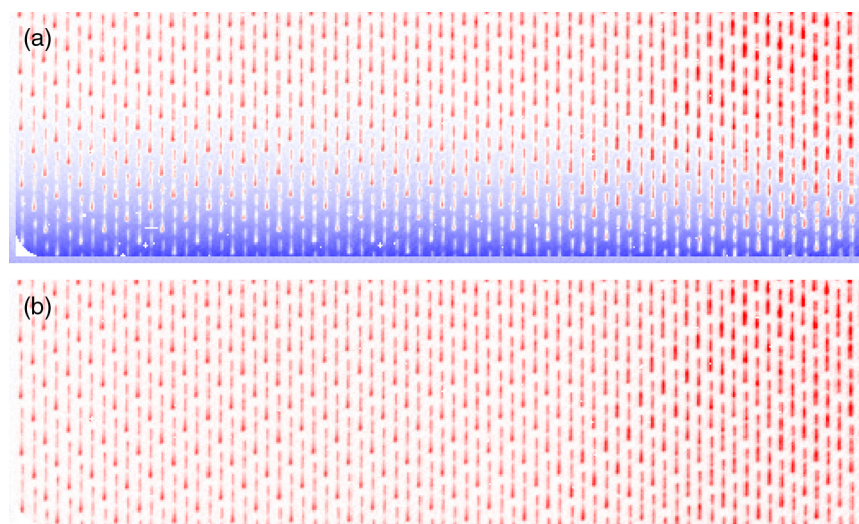
### 3.2 Nonlinearity and Saturation

As CHARIS’s H2RG approaches saturation, its response becomes nonlinear. The nonlinearity sets in gradually before the pixel’s response drops sharply to zero. The H2RG also suffers from bleeding into adjacent pixels. After a pixel saturates, its four nearest neighbors see an immediate and compensating increase in their count rates, while its four next-nearest neighbors (the nearest along a diagonal) see a smaller, but still significant, increase. Pixels that do not adjoin the saturated pixel show no significant increase in their count rates until an adjacent pixel saturates.

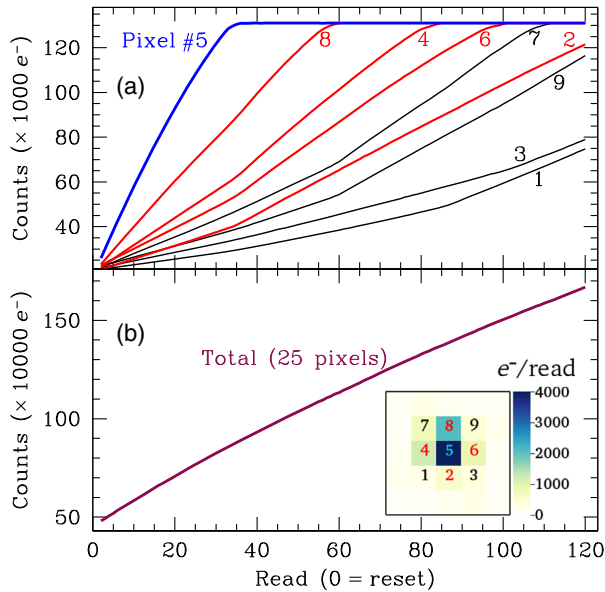
Figure 3 shows the detector’s response to saturation in a cut-out of a 120-read ramp in which the most strongly illuminated pixels saturate in  $\sim 30$  reads. The lines in Fig. 3(a) are labeled with the pixel number, and an inset in Fig. 3(b) provides the key. The increase in count rates in the peripheral pixels closely approximates the lost counts from the saturated pixel(s). The total instantaneous count rate in a  $5 \times 5$  box decreases by just  $\sim 35\%$  between the first few and the last few reads, despite the fact that saturated pixels account for  $\sim 70\%$  of the photons in the initial reads.

The H2RG’s saturation behavior means that the saturation of one pixel immediately corrupts the count rates of its eight nearest neighbors but can be neglected for pixels that are farther away. We fit for each pixel’s count rate using only the reads for which neither the pixel in question nor any of its immediate neighbors have saturated.

We measure our H2RG’s nonlinearity using a series of long exposures: a 1000-read ramp taken with the detector almost uniformly illuminated and 16 120- to 160-read ramps with the detector sparsely illuminated by monochromatic light passing through the lenslets. In both cases, we fit for each pixel’s



**Fig. 2** The lower-left corner of a 17-read ramp taken in CHARIS’s broadband, low-resolution mode (a) before and (b) after removing the exponential decay of the reference voltage in the first read. Fitting the reference voltage requires a negligible fraction of the information contained in the first read. The read may then be used normally to increase the integration time on-source and reduce read noise. The microspectra are visible as short vertical lines with gaps corresponding to the atmospheric absorption bands between the *J*- and *H*-bands and between the *H*- and *K*-bands.



**Fig. 3** Leakage of electrons into neighboring pixels upon saturation. (a) The actual counts (assuming a gain of 2) as the individual pixels shown in the inset saturate. Neighboring pixels absorb the extra electrons upon saturation. (b) In spite of the saturation of pixels 4, 5, 6, 7, and 8 (accounting for nearly 70% of the counts rate in the initial few reads), the total count rate in the  $5 \times 5$  pixel box decreases by just  $\sim 35\%$  between the first few and the last few reads.

count rate using only the first  $\sim 5\%$  of the reads. We then compare the actual counts at subsequent reads with the expected count rates from the initial reads assuming perfect linearity.

Figure 4 shows the density of points in actual versus predicted counts on the detector. The density is independently normalized at each count rate ( $x$ -coordinate) to limit dynamic range,

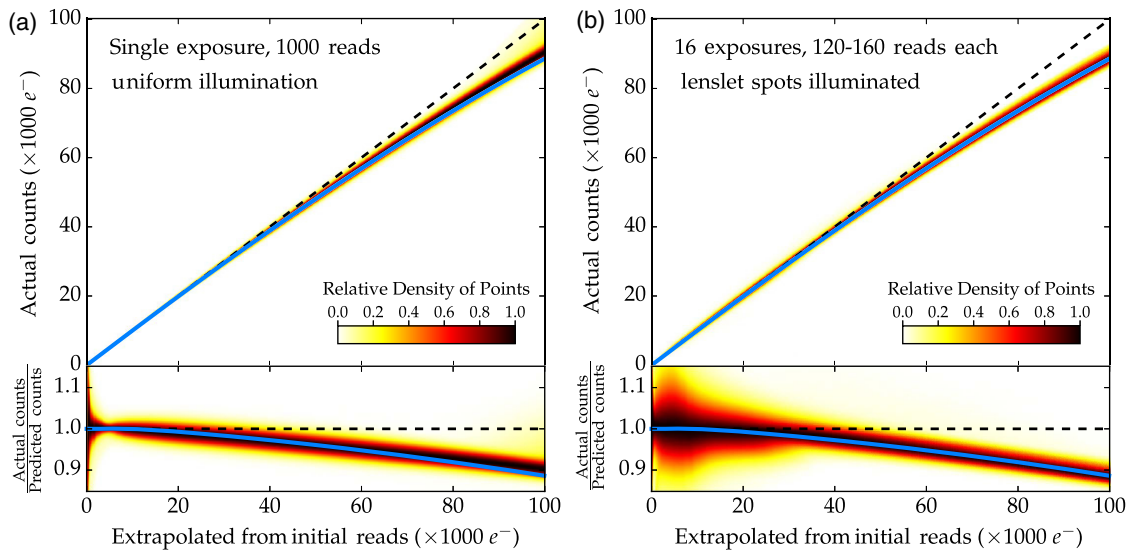
and there are several billion points in each figure. A perfectly linear detector should be nearly symmetric about the dashed line  $y = x$  (biases from read and photon noise are  $< 1\%$  at these count rates). Our H2RG falls below the line, indicating a loss of sensitivity as the pixels approach well capacity. The solid blue line shows our adopted nonlinear response: linear up to  $10,000 e^-$ , continuously matched to a cubic fit up to saturation. Our method is distinct from Finger et al.,<sup>36</sup> who only use unsaturated reads to fit a ramp but assume the pixel response to remain linear.

Different ramps suggest nonlinear responses that differ by  $\sim 1\%$  to  $2\%$  near saturation. We make no attempt to explain or account for this in the data, but we do allow the user to add a fixed fraction of the count rate as an uncertainty. This error term also accounts for imperfect modeling of the microspectra, and an appropriate value (typically  $\sim 5\%$ ) gives a  $\chi^2$  per pixel of order unity after fitting all of the 2-D microspectra. This 5% uncertainty is significantly larger than the  $\sim 1\%$  uncertainty in the nonlinear response.

Our adopted pixel response is linear up to  $10000 e^-$  or  $\sim 10\%$  of well capacity. For pixels that remain below this value, we, therefore, use the UTR fit described in the first paragraphs of Sec. 3. We individually fit pixels that exceed this threshold. With a fixed nonlinear pixel response, the nonlinear fit has two free parameters: the initial count rate and the reset value. The best-fit count rate  $x$  minimizes

$$\chi^2[x, b] = \sum_j \frac{(F_j - f[x \cdot j] + b)^2}{\sigma^2}, \quad (6)$$

where the sum is over the reads  $j$ ,  $b$  is the reset value, and we neglect photon noise. For a linear detector,  $f[x \cdot j] = x \cdot j$  with  $x$  in units of counts/read and the best-fit  $x$  is given by the usual UTR weights [Eq. (3)]. We exclude from the fit all pixels with



**Fig. 4** The nonlinear response of CHARIS's detector as measured (a) in a 1000-read ramp taken before the full baffling was installed and (b) in a series of 120- to 160-read ramps taken with the detector sparsely illuminated by monochromatic light passing through the lenslet array. The color shows the density of measurements normalized to the maximum density at a given extrapolated number of counts ( $x$ -coordinate); each plot contains data from several  $10^9$  points. The black-dashed lines show a perfectly linear response, while the blue lines show our fiducial fit to the nonlinear response. The frames differ slightly in the nonlinear response, at the level of  $\sim 1\%$  to  $2\%$ , as the pixels approach saturation ( $\sim 100,000 e^-$  above reset). Our fit assumes that the pixels have a linear response of up to  $10,000 e^-$  ( $\sim 10\%$  of well capacity).



a predicted count rate of more than  $>10^5 e^-$  above reset or with  $>60,000$  raw counts ( $1.2 \times 10^5 e^-$  at a gain of 2; values above 65,535 raw counts cannot be represented by unsigned 16-bit integers).

In our case,  $f[x \cdot j]$  is a nonlinear function of  $x$ , and the count rate may no longer be obtained as the weighted sum of the reads with the coefficients from Eq. (3). However, at a fixed count rate, finding the best-fit reset value  $b$  remains a linear problem, and the best  $\chi^2$  at this count rate is trivial to compute. We use this fact to reduce the 2-D nonlinear minimization in  $x$  and  $b$  to a 1-D nonlinear minimization in  $x$  (the coefficients of the cubic function define the nonlinear response and remain fixed). We begin with a guess from the UTR count rate and locally fit a parabola to  $\chi^2[x]$ ; the vertex of the parabola is our next guess for the count rate. As with our procedure to fit for the initial exponential decay of the reference voltage, this is equivalent to Newton–Raphson iteration on  $d\chi^2/dx$ . The algorithm converges in only a few steps and requires a negligible amount of computation for a typical ramp.

### 3.3 Read Noise Properties

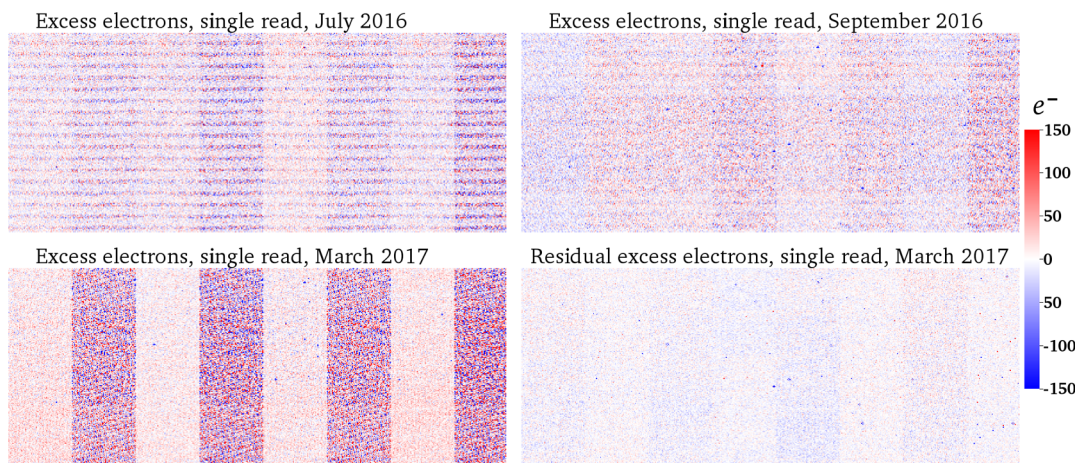
CHARIS’s H2RG detector has  $1/f$  read noise that is correlated among the readout channels; this behavior is typical of H2RGs.<sup>33</sup> It may be suppressed by fitting out power on various timescales using interspersed reference pixels,<sup>33</sup> using (nearly) unilluminated light-sensitive pixels<sup>42</sup> or using different weightings of the reference pixels at the detector edges.<sup>43</sup> In CHARIS, there is an additional component of read noise that is largely shared by alternating readout channels. We read out our detector using 32 readout channels, each  $2048 \times 64$  pixels in size. The even and odd channels each have their own shared component of read noise: removing a single scaled template from all channels

achieves only  $\sim$ half the noise suppression of using different templates for the even and odd channels.

The power spectrum of CHARIS’s read noise spikes at a range of frequencies in the 1- to 10-kHz range, most of which do not match any known frequencies of the system. The noise is also highly variable with time, both in amplitude and in its power spectrum. In lab tests during CHARIS’s assembly, the shared read noise was comparable in power to the independent read noise in each channel. On Subaru’s Nasmyth platform, the correlated component of the read noise has gotten much worse; its variance can be more than 10 times that of the independent read noise depending on the date and the readout channel. We are currently investigating this source of noise and attempting to fix it in hardware.

Figure 5 shows three representative realizations of CHARIS read noise on three widely separated dates. In all cases, we have used a long ramp to predict the count rate at the second read of the ramp and then used the difference between the actual and expected counts in the second read as a measure of the read noise. The noise in a CDS image (a difference of two reads) would be  $\sqrt{2}$  times the noise shown.

Figure 5 shows that the read noise on Subaru’s Nasmyth platform is a serious problem, with CDS-equivalent read noise  $\gtrsim 100 e^-$  in some channels at some times and that the read noise properties vary with time as CHARIS and neighboring instruments are moved and reconfigured. The lower-right panel of Fig. 5 demonstrates how much of this read noise is shared among alternating readout channels. In this panel, we have constructed two read noise templates: one for the even channels and a separate one for the odd channels. We then rescaled the appropriate template to each channel and removed it. This procedure reduced the read noise by a channel-dependent factor of 1.3 to 5.8. If there were no correlated read noise,



**Fig. 5** Sample realizations of the read noise on three different dates. The read noise is computed as the difference between the actual number of counts at a single read and the expected number of counts based on the other reads in the ramp. The July 2016, September 2016, and March 2017 images have root-mean-square noise of about  $33 e^-$ ,  $31 e^-$ , and  $53 e^-$ , respectively. The strong banding in July 2016 (much fainter but still visible in September) has a modulation at a frequency of 120 Hz. In March of 2017, the read noise shows a dramatic difference in amplitude between the even and odd readout channels. Much of the noise in all of these images is in two patterns, one common to all of the even channels and one to all of the odd channels. The lower-left panel shows the same March 2017 read as the lower-left panel, but with a scaled pattern subtracted from the even readout channels and a different scaled pattern subtracted from the odd channels. The read noise falls by a channel-dependent factor of 1.3 to 5.8 (the variance falls by a factor of  $\sim 2$  to 30), when this correlated read noise is removed, and the overall root-mean-square falls from about  $53 e^-$  to about  $14 e^-$ . We discuss the removal of correlated read noise from CHARIS ramps in Sec. 5.3.

this procedure would only be expected to reduce the variance by  $\sim 1/16$  or the noise by  $\sim 3\%$  (since 16 channels were used to construct each template).

Real CHARIS ramps are packed with microspectra, preventing us from simply removing the correlated read noise as described above. However, the combination of our  $\chi^2$  method of fitting the microspectra (Sec. 5.2) and the generous spacing of the spectra on the detector does enable us to achieve effective read noise suppression. We discuss this in more detail in Sec. 5.3.

#### 4 Calibrations for Cube Extraction using Monochromatic Flatfields

A CHARIS image consists of microspectra arrayed in a grid on the detector (see Fig. 2); these must be extracted into a data cube. Each microspectrum is the integral of the monochromatic lenslet spots over the spectrum of light that the lenslet sees, convolved with the pixel response function. Extracting the source spectrum requires knowing the pixel locations where each wavelength of light falls for each lenslet; this is the wavelength solution. Optimal extraction and  $\chi^2$  extraction, the two main algorithms included in the CHARIS data reduction software, also require knowledge of the monochromatic lenslet spots (the lenslet PSF). In the rest of this paper, we will use the terms lenslet PSF and PSFlet interchangeably.

To both derive the wavelength solution and measure the wavelength-dependent PSFlets, we inject a supercontinuum source with a narrowband tunable filter into an integrating sphere to uniformly illuminate CHARIS's lenslet array with monochromatic light. The tunable filter has a width of 5 nm, for a spectral resolution of  $\sim 300$  at  $1.6 \mu\text{m}$ , well in excess of CHARIS's resolving power in its high spectral resolution mode. Such a calibration strategy would need to be revised for an instrument with higher spectral resolution, though a lamp with well-spaced emission lines might serve as an effective substitute for our tunable filter. We have also integrated three narrowband filters, one each in the  $J$ ,  $H$ , and  $K$  bands, into SCEXAO's optics. With one of these filters in place, we may uniformly illuminate CHARIS's lenslet array with any infrared-bright lamp or even the twilight sky.

We break the CHARIS calibration procedure into two steps. We first use our supercontinuum source and tunable filter to gradually step through the wavelength. This enables us to measure the locations of the lenslet PSFs on the detector and to

derive the wavelength solution. We also use these images to extract the position-dependent lenslet PSFs. This first step, a full sequence of calibration images, is rarely (if ever) repeated. As of publication, the pipeline includes calibration products derived from a July 2016 calibration sequence. A new calibration sequence offers negligible improvements even one year and several cooling cycles later. The second step is to use a narrow-band flat to derive a correction to the wavelength solution once per night. This section describes the process in detail and summarizes the final calibration files that it produces.

The typical user of CHARIS will not need to derive the full wavelength solution or the monochromatic lenslet PSFs; we have calculated these and distribute them with the source code. The user will, however, need to build the appropriate calibration files for a given observing mode from a single narrow-band flat as thermal cycles of the instrument induce small shifts of the microspectra on the detector. We include the script `buildcal` for this purpose. It takes as input the raw reads from a single narrowband flat and writes all of the necessary calibration files to the directory from which it was run.

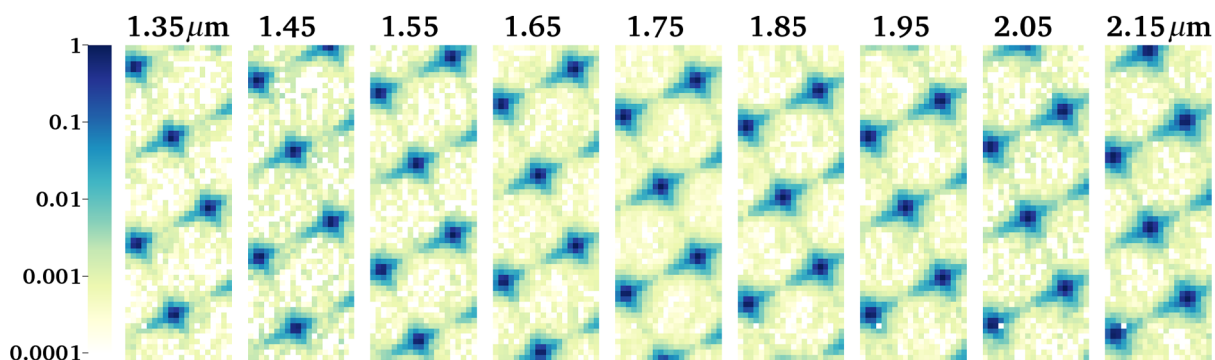
##### 4.1 Full Calibration Sequences

A full calibration sequence consists of connecting our supercontinuum source to a tunable filter, coupling it to an integrating sphere, and injecting the light into CHARIS to illuminate the lenslet array. We then step through the wavelength with a step size comparable to or slightly larger than CHARIS's spectral resolution. This yields a series of about 10 to 15 images per observing mode with well-separated lenslet PSFs. Figure 6 shows these spectrally unresolved flatfields from  $1.35$  to  $2.15 \mu\text{m}$  for a small subregion of the detector.

We use these calibration sequences to measure the PSFlet locations for each lenslet as a function of wavelength. We assume a cubic polynomial mapping between integer lenslet coordinates  $(i, j)$  and floating point pixel coordinates  $(x_{ij}[\lambda], y_{ij}[\lambda])$ , with the polynomial coefficients being functions of wavelength

$$\begin{Bmatrix} x_{ij}[\lambda] \\ y_{ij}[\lambda] \end{Bmatrix} = \sum_{m=0}^3 \sum_{n=0}^{3-m} \begin{Bmatrix} a_{nm}[\lambda] \\ b_{nm}[\lambda] \end{Bmatrix} i^m j^n. \quad (7)$$

We first lightly smooth our monochromatic images with a narrow Gaussian; we use the known lenslet pitch and rotation



**Fig. 6** A partial calibration sequence through CHARIS's broadband mode showing the motion of the lenslet PSFs, or PSFlets, in the dispersion direction (toward the bottom of the page). Only a small region of the detector is shown ( $20 \times 60$  pixels; a microspectrum is just over 30-pixels long), and the intensity is plotted on a logarithmic scale normalized to the peak of the brightest PSFlet in the region shown. The diagonal spikes are due to diffraction from the square lenslets.



as our initial guesses for the linear coefficients of the polynomial. We then maximize the sum of the interpolated intensities at the lenslet spot locations by adjusting the coefficients. Once we have derived these coefficients for one wavelength, we estimate the offset in the dispersion direction for the next wavelength using a grid search. We combine this offset with the polynomial coefficients from the previous step to form the initial guess for this optimization. We proceed to derive the cubic polynomial transformation from lenslet to detector coordinates for all wavelength steps.

For any wavelength of interest, we can now compute all of the coefficients of the lenslet-detector transformation polynomial (and hence the full wavelength solution) by fitting a cubic polynomial as a function of  $\log \lambda$  to each coefficient. We have derived these full wavelength solutions for each CHARIS observing mode using calibration sequences taken in July 2016 and distribute them as part of the CHARIS software package.

Our next step is to reconstruct the lenslet PSFs. For wavelengths  $\lesssim 2 \mu\text{m}$ , these are undersampled by CHARIS's H2RG. We take an approach very similar to Ref. 44 for *Hubble* and to Refs. 45 and 46 for GPI. By deriving the wavelength solution, we already have the location of each lenslet PSF's centroid and can place it on an oversampled grid. The lenslet PSFs are not spaced by an integer number of pixels nor by a ratio of small integers; as a result, they populate an oversampled PSFlet reasonably densely. We iteratively construct a slightly smoothed PSFlet and fit for each PSFlet's normalization to refine the smoothed template. Finally, we deconvolve with the smoothing kernel using the Richardson–Lucy algorithm<sup>47</sup> to produce our final oversampled PSFlets. We measure these

oversampled PSFlets at about 10 wavelengths in each observing mode.

Figure 7 shows our resulting oversampled PSFlets in 25 sub-regions of the detector for  $1.55\text{-}\mu\text{m}$  monochromatic light. We use bilinear interpolation to estimate the PSFlet between the centers of regions and assume that the PSFlets remain constant from the centers of the outer regions to the edges of the detector. The convolution and deconvolution are not completely equivalent because some subpixel offsets are sampled more than others; this could lead to a systematic underestimation of the PSFlet width. Random errors in the PSFlet centers would push in the other direction, leading to a systematic overestimation of PSFlet width. An inspection of the residuals after our  $\chi^2$  fitting of the microspectra (Sec. 5.2) indicates that these errors are negligible. Our use of only 25 images to represent PSFlet variation over the detector (motivated by the need to average over a large number of lenslets) is probably a bigger source of error.

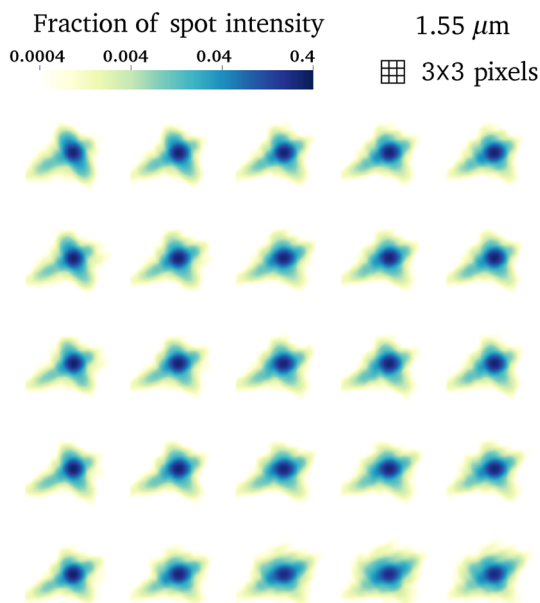
As of publication, the pipeline includes one set of calibration files derived from calibration sequences taken in July 2016. We have been unable to conclusively measure any differences in the dispersion, in the nonlinear part of the wavelength solution, or in the PSFlet shapes over several calibration sequences and cooling cycles. Wavelength solutions based on more recent calibration sequences have not improved the extracted data cubes so long as a contemporaneous narrowband flat is available (see Sec. 4.2 for details). We will add updated calibration files to the pipeline if and when they prove necessary.

## 4.2 Narrowband Flats

We use the long calibration sequences described above to measure the lenslet PSFs as a function of both wavelength and position and to compute the wavelength solution. This wavelength solution changes as CHARIS thermally cycles and as it is craned onto and off of its bench. Small shifts due to motion of the optics or lenslet array change the position and orientation of the wavelength solution but do not have a measurable impact on its nonlinear component. We use a single narrowband flat for each night to compute these shifts and adjust the wavelength solution accordingly, assuming that its nonlinear component remains fixed.

Our strategy for these narrowband flatfield images is to uniformly illuminate CHARIS's lenslet array using the halogen flatfield lamp in front of AO188, the system optically upstream of SCEXAO. One of three narrowband filters within SCEXAO creates a flatfield image that CHARIS cannot resolve spectroscopically. These filters, at 10-nm width, are somewhat broader than our tunable source ( $\sim 5 \text{ nm}$ ); the 1200-nm *J*-band light ( $R \sim 120$ ) is marginally resolved by CHARIS in its *J*-band mode ( $R \sim 80$ ). All filters are completely unresolved in CHARIS's broadband mode.

We process a narrowband flat using the same procedure as for the full calibration sequence above, except that we use the existing wavelength solution as a starting guess for the spot locations. We perform a grid search to find the approximate offset and then iterate using Powell's method within the minimization function of `scipy.optimize`. We keep the offset and linear terms of this solution and replace the higher-order terms with their values in the main wavelength solution. These refinements to the wavelength solution are all that we extract from the narrowband flats; they are the only aspects of CHARIS calibration data that are unstable from run to run. Because CHARIS sits on Subaru's Nasmyth platform and because we designed the



**Fig. 7** Oversampled  $1.55\text{-}\mu\text{m}$  PSFlets reconstructed over 25 sub-regions of the detector. The observed PSFlets are resampled by the integer pixel grid of the detector; a  $3 \times 3$  pixel grid in the upper-right shows the pixel size. The PSFlets are normalized to unit intensity after resampling. The oversampled PSFlets allow us to construct the pixellated microspectra corresponding to monochromatic and broadband light incident on the lenslet array. The crosses are diffraction spikes from the square lenslets, while the shape variations show the focus changing across the field.

imaging relay to be thermally stable (with aluminum mirrors and an aluminum bench), the location of the PSFlets is very stable within a night. Project 1640 and GPI are both mounted at the Cassegrain focuses of their respective telescopes; their PSFlets shift due to flexure as the telescope elevation angle changes.<sup>48,49</sup>

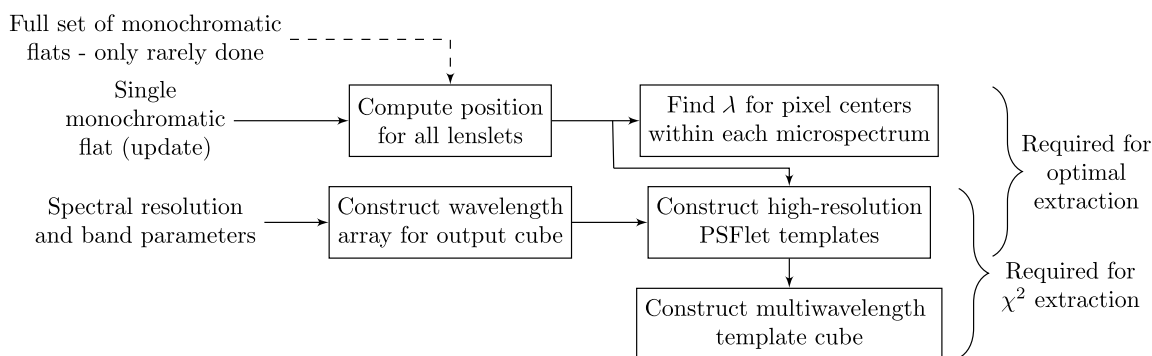
We include the script `buildcal` with the CHARIS software for the purpose of building all of the necessary calibration files. The user runs `buildcal` with the raw file for the narrowband flat as a command-line argument. The software will then compute the calibrations and write all of the files to the directory from which `buildcal` was run. Some of these files are specific to optimal extraction and  $\chi^2$  extraction, the two main spectral extraction techniques used by the pipeline. These files are described together with the methods themselves in Secs. 5.1 and 5.2. A summary of the wavelength calibration steps and required products for each extraction technique is shown in Fig. 8.

### 4.3 Masking and Flatfielding

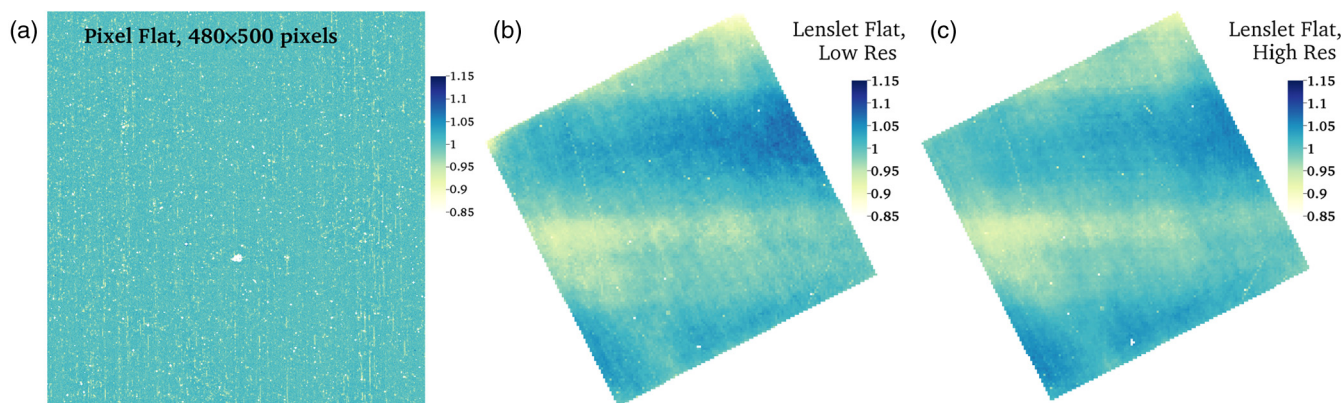
Flatfielding an ordinary image divides by a single sensitivity for each pixel; that sensitivity is a combination of illumination by the optics and the quantum efficiency of the pixel. In an IFS like CHARIS, these two components of the flatfield separate into a detector flat that measures the pixel-to-pixel sensitivity differences and a lenslet flat that measures differences in the illumination and transmission of the various lenslets.

We construct the pixel flat from early ramps taken before CHARIS was effectively baffled, when the detector was (relatively) uniformly illuminated. This flatfield does include artifacts from the nonuniform illumination. We therefore apply a high-pass filter, preserving pixel-to-pixel variations on the scale of the microspectra but removing the slow variations in illumination across the chip. Individual CHARIS microspectra are  $\sim 30 \times 6$  pixels in size, and our high-pass filter is a 2-D Gaussian with  $\sigma = 10$  pixels (FWHM 23.5 pixels) and removes power on scales significantly larger than this. We use prebaffling images, together with later images where the background count rate is very low, to identify bad pixels. These are either hot, with a very high dark current, or they respond to light much more weakly than their neighbors. Later images effectively identify the hot and warm pixels, while our method of constructing the pixel flat identifies pixels that are not light sensitive. We flag a pixel as “bad” if it has a dark current 15 sigma above the read noise of its neighbors in a series of long-exposure dark frames (i.e., above  $\sim 0.4 \text{ e}^- \text{ s}^{-1}$ ) or if its sensitivity in our high-pass-filtered pixel flat is below 80%. We flag 0.6% of pixels as bad in this way; they are clustered in groups large and small across the detector.

Our second step is to construct a lenslet flat. For this step, we return to our full calibration sequences from which we derived a wavelength solution and measured the high-resolution lenslet



**Fig. 8** Wavelength calibration steps and required products for the two main reduction methods. A single monochromatic flat for each observing mode is the only data required by the CHARIS user to update all of these products and proceed to cube extraction. These steps are implemented in the script `buildcal`.



**Fig. 9** (a) A subregion of the pixel-based flatfield constructed before effective baffling was installed. We used a high-pass filter to remove the illumination pattern. (b and c) Lenslet flatfield images constructed by comparing the monochromatic spot pattern expected for perfectly uniform throughput and the spot patterns actually observed. The flatfields shown are averaged over wavelength and are very nearly consistent between the low- and high-resolution modes.

PSFs. Having extracted these, we reconstruct the monochromatic spot pattern expected for a uniformly sensitive detector and a uniformly illuminated lenslet array. We then scale this ideal spot pattern by the pixel flat described above and determine the best-fit lenslet-by-lenslet amplitude of the PSFlets. Because the spot pattern is monochromatic, the spots are very well separated and cross-talk among neighboring lenslets may be safely ignored (see Fig. 6).

We thus obtain a lenslet flat for each wavelength in our sequence. The flats are consistent with one another within the observing mode, as is expected for illumination of the field by nearly all reflective optics. This also indicates that the filters, which lie immediately behind the lenslet array, have transmission curves that are nearly spatially uniform. We median-combine the flats to create our lenslet flat for each observing mode. We have also verified that these lenslet flats are consistent with those that we would derive from our narrowband flatfield images, described in Sec. 5.

Figure 9 shows the lenslet flat and a subregion of the pixel flat. The pixel flat is the same for all observing modes (it was measured using white-light while the detector was poorly baffled), while there is, in principle, a different lenslet flat for each observing mode. In practice, the lenslet flat is nearly the same in all modes due the almost exclusive use of reflective optics throughout CHARIS and the AO systems. The pixel sensitivity generally varies by just a few percent from one pixel to the next. The lenslet flat, apart from a few poorly illuminated (or effectively opaque) lenslets, has typical variations of  $\sim 20\%$  across the array.

## 5 Cube Extraction

Once we have constructed all of the calibration materials, extracting a data cube is relatively straightforward. CHARIS includes two main techniques to perform the extraction: optimal extraction<sup>50</sup> and  $\chi^2$  extraction. We describe our implementation of each below. The CHARIS pipeline also implements a simple aperture photometry-based extraction. However, this algorithm is unable to account for bad or noisy pixels and has no advantage, either in performance or run time, over optimal extraction.

### 5.1 Optimal Extraction

Optimal extraction<sup>50</sup> computes the spectral intensity at each wavelength (i.e., each row of pixels perpendicular to the dispersion direction) using both the measured shape of the line-spread function and the pixel-specific measurement errors to weight the pixels. This extraction method has long been the standard approach for spectrographs.<sup>51–53</sup> We implement optimal extraction for CHARIS assuming Gaussian profiles with a wavelength- and lenslet-dependent width that we measure from our high-resolution PSFlets (Fig. 7). This is equivalent to computing the spectral intensity as the normalization of a 1-D Gaussian of known position, width, and unit area.

We compute the position of each lenslet's microspectrum as part of the calibration process, but the wavelengths that correspond to integer pixels along the spectrum differ. Each microspectrum is  $\sim 30$  pixels long in the dispersion direction. We obtain  $\sim 30$   $(\lambda, \sigma, x, y)$  quadruples for each lenslet, where the positions are integers in the dispersion direction  $y$  and floating point numbers in the perpendicular direction  $x$  (which runs along the center of the microspectrum). The wavelength-dependent width  $\sigma$  of the microspectrum is given by computing the second moment of the flux along a line passing through

the center of the corresponding PSFlet (Fig. 7). Because the PSFlets are not circularly symmetric, the actual profile will include variable contributions from the diffraction spikes at other wavelengths and will differ depending on a lenslet's spectrum. Optimal extraction takes the  $(\lambda, \sigma, x, y)$  quadruples for each lenslet (which are computed and saved as part of the calibration step) and uses a weighted sum to calculate a corresponding spectral intensity.

Optimal extraction returns the spectral intensity at the wavelengths corresponding to a given microspectrum's sampling on the detector. Each microspectrum has its own exact spectral resolution and subpixel sampling and is, therefore, defined on its own native wavelength array. It is possible to extract the data without interpolating onto a common wavelength array, but it is difficult to visualize and manipulate such data. CHARIS's software, therefore, returns a cube in which these microspectra are interpolated onto a common wavelength array at the spectral resolution requested by the user. The resulting data cube couples neighboring wavelengths because of this interpolation and because the extracted microspectra are convolved with the line-spread function (a PSFlet's extent along the dispersion direction of a microspectrum).

The output of optimal extraction is a pair of data cubes: one for the spectral intensities and one for their errors. The code could easily be modified to return cubes at CHARIS's native wavelength sampling (which differs for each lenslet). In that case, the software would return three cubes: one for the spectral intensities, one for their errors, and one for the native wavelengths.

The CHARIS software also has the ability to perform a naïve aperture extraction using unit weights perpendicular to the dispersion direction. This approach is currently used by SPHERE and GPI.<sup>22,54,55</sup> For CHARIS, aperture extraction produces cubes with slightly more noise (depending on the aperture) and lacks an ability to treat errors and bad pixels: bad or noisy pixels must be replaced with guesses from their neighbors. We do not attempt to replace bad pixels in this case but simply set them to zero. A more careful implementation of aperture photometry would produce a result no better than that from optimal extraction and would save a negligible amount of computational cost.

### 5.2 $\chi^2$ Extraction

CHARIS's software implements a  $\chi^2$ -based extraction, fitting every microspectrum with a linear combination of narrowband spots. Once the narrowband spot templates have been computed, it is relatively straightforward to extract the spectrum, i.e., the coefficients of their best-fit linear combination. Computing the templates themselves, however, is more subtle. The spectra on the detector are the spectra seen by the lenslets and convolved with both the lenslet PSFs and the pixel response function. An attempt to use monochromatic PSFlets to fit the microspectra will suffer from the fact that no real spectrum can be represented as the sum of delta functions. Attempting to overcome this by extracting a spectrum at a resolution significantly higher than CHARIS's native resolution would cause severe problems with aliasing and covariance among neighboring wavelengths. Our solution is to fit the microspectrum of each lenslet as a series of top-hat spectra in units of  $I_\lambda$ , with the spectral resolution of this sampling chosen to be comparable to, or slightly higher than, CHARIS's intrinsic spectral resolution.



We construct a spot diagram for a narrow spectral range using the oversampled lenslet PSFs, such as those shown in Fig. 7. We use 10 monochromatic spots to construct each narrowband spot, first scaling each monochromatic spot by the atmospheric and filter transmission appropriate to the very narrow range of wavelengths that it covers. In this way, our narrowband spectra are what we would expect for an astrophysical source with a perfectly flat  $I_\nu$ , average atmospheric transmission, the filter for the given observing mode, and achromatic optics in the rest of the system. The correction is far from perfect, but it does mitigate the effect of the atmosphere and filter on our recovered microspectra.

Our calibration routine produces and saves a series of narrow, but not quite monochromatic, arrays of spots stepping through wavelength and built as described above. Together, these spots cover the full wavelength array of a given observing mode with no gaps and approximately correct for the chromatic throughput of the atmosphere and filter. Figure 10 shows these narrowband spot diagrams. They are broader than those as shown in Fig. 6, but only slightly.

Finally, we produce one last set of narrowband spots where we retain an oversampling by a factor of 5 in the direction perpendicular to the dispersion. This allows us to use cross correlation to find the appropriate offset for an individual CHARIS ramp even if a calibration data set is not available from that night (or if it proves a relatively poor match). This takes a significant amount of disk space ( $\sim 2$  GB per calibration set) but enables the spectra to be located to  $\sim 0.01$  pixels and more accurately fit by our extraction routine. We do not oversample in the dispersion direction because we currently have no way of getting a sufficiently accurate wavelength solution for an individual image without a narrowband calibration flat.

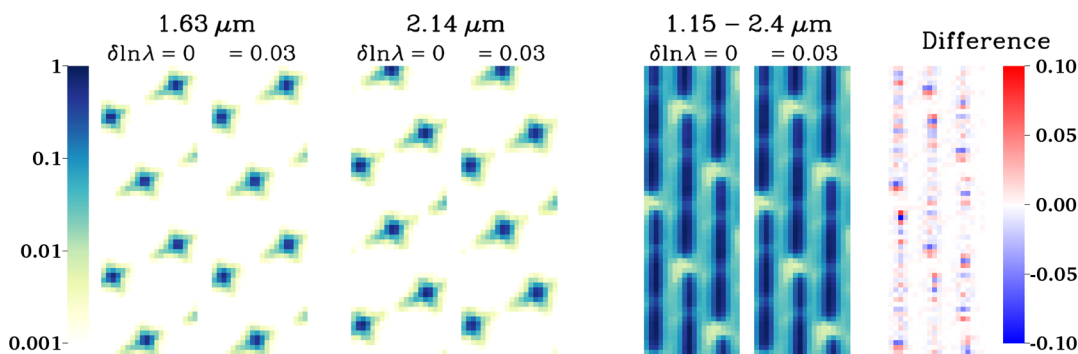
Once all of the narrowband templates shown in Fig. 10 have been computed and saved,  $\chi^2$  spectral extraction is relatively straightforward. We begin by cutting out a 7-pixel wide,  $\sim 35$ -pixel-long rectangle around each microspectrum. We then fit each 2-D cutout with 20 to 25 narrowband spots, minimizing the squared residuals weighted by the pixels' inverse variance. We use the singular value decomposition for this purpose, implementing it ourselves in Cython to enable lenslet-by-lenslet parallelization using OpenMP. This approach naturally includes the errors on individual pixels and masks hot pixels, and

it returns the spectral covariance matrix for each lenslet. It also avoids any interpolation onto a common wavelength array as was necessary for optimal extraction. A similar approach has been implemented for GPI,<sup>46,45</sup> but measurements of GPI's wavelength-dependent lenslet PSFs were insufficient for the algorithm to perform well.

Figure 11 shows the performance of our  $\chi^2$  extraction on the microspectra in two regions of a low-resolution image: one far from the star where read noise is important (a) and the other in a bright speckle where it is negligible (b). The residuals from systematic errors in the PSFlet models are typically  $\sim 5\%$  of the intensity. As we discuss below in Sec. 5.3, our models (center panels) include the correlated component of the read noise and the undispersed background. This is visible in the top panel, where the left few columns of microspectra are in a noisy readout channel while the rightmost columns are in a much cleaner channel.

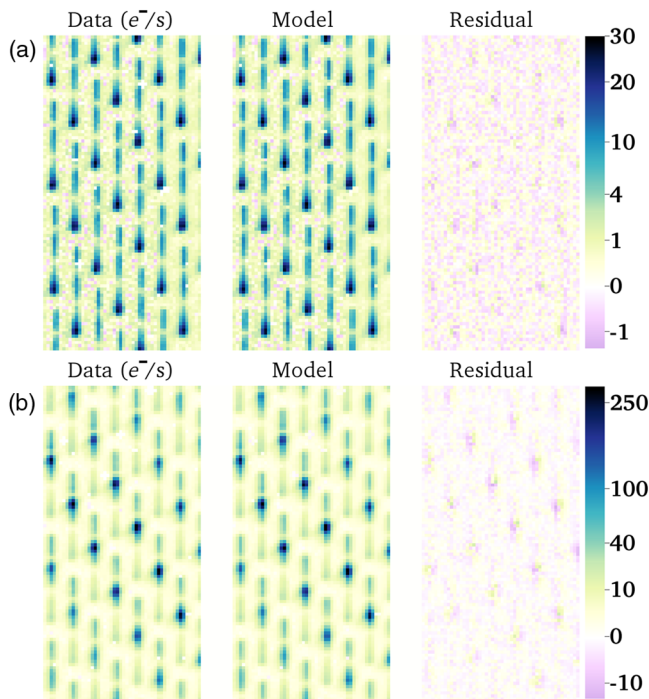
As an optional feature, the software can use the PSFlet template file that is oversampled in the direction perpendicular to the dispersion. It uses cross correlation over  $32 \times 32$  subregions of the detector to fit a position-dependent subpixel shift in the locations of the spectra. We adopt a range of prospective shifts spaced by 0.2 pixels, compute the cross correlation in each case, and then fit a parabola to the three cross-correlation values nearest their minimum to obtain the exact subpixel offset. We then use bilinear interpolation to compute an offset for each lenslet from the  $32 \times 32$  cross-correlation offsets and interpolate the oversampled PSFlet templates onto the appropriate pixel sampling. This approach typically produces an offset of no more than a few tenths of a pixel but noticeably improves the residuals in the 2-D microspectra. The fit shown in Fig. 11 includes this subpixel offset.

A  $\chi^2$ -based extraction has several features that differentiate it from optimal extraction and aperture photometry on the microspectra. Because we fit the entire 2-D spectrum, it is trivial to also fit for and remove an undispersed background that is uniform over the microspectrum. For microspectra that straddle two readout channels, we allow for the undispersed background to have different values in the two channels. This fitting of an undispersed background is an optional setting in the software's configuration.



**Fig. 10** An illustration of the calibration products for  $\chi^2$  extraction. The panels labeled  $\delta \ln \lambda = 0$  use monochromatic spots, while the panels labeled with  $\delta \ln \lambda = 0.03$  account for the finite bandwidth separating the spectral measurements and include wavelength-dependent atmospheric transmission. The lenslet spots with  $\delta \ln \lambda = 0.03$  are very slightly (almost imperceptibly) broader than those with  $\delta \ln \lambda = 0$ . When modeling the microspectra of a source with constant  $I_\nu$ , the difference (right panel) between using monochromatic spots (third from right) and spots of finite bandwidth (second from right) is a high-frequency pattern with an amplitude of around 10% of the maximum intensity, peaking at the edges of the bandpass.





**Fig. 11** The performance of our  $\chi^2$  extraction algorithm on low-resolution microspectra in two regions of an image: (a) one far from the star where read noise is significant and (b) one in a bright speckle where it is negligible. We have used a square root stretch to more clearly show residuals. Our model includes the undispersed background and the correlated component of the read noise, computed as described in Sec. 5.3. In the absence of read noise, the residuals are typically  $\sim 5\%$  of the input data.

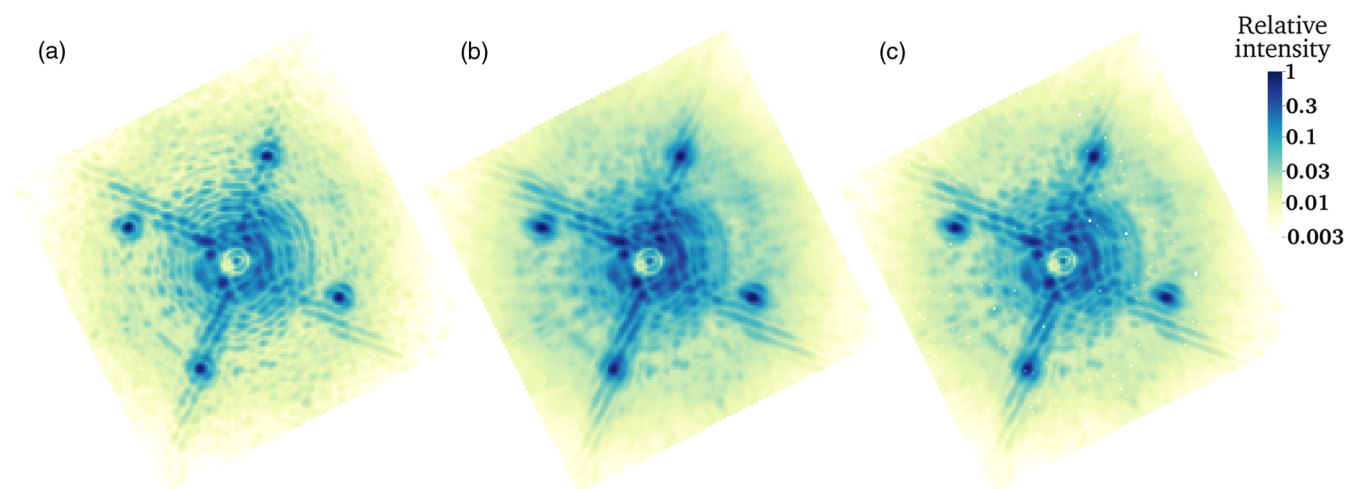
An important feature of the  $\chi^2$  extraction is that it automatically attempts to extract the intrinsic source spectrum; it performs a deconvolution with the instrumental line-spread function. This deconvolution results in a negative covariance between neighboring spectral bins and a superficially noisier

cube than that produced by optimal extraction. A small amount of smoothing in the spectral dimension, i.e., a reapplication of the convolution with the line-spread function, removes this extra apparent noise. The effect of the line-spread function is apparent in a single slice through a data cube, as shown in Fig. 12. With optimal extraction, the speckles are radially extended due to contributions from a range of wavelengths. This arises both from the line-spread function and from the fact that we interpolate all of our microspectra (each of which has a different native wavelength sampling) onto a common wavelength array. The speckles in a  $\chi^2$ -extracted cube show a negligible radial extent beyond that of the monochromatic PSF incident on the lenslet array.

A  $\chi^2$  extraction naturally enables the removal of CHARIS's spectral cross talk. In the calibration step, we measured the lenslet PSFs (shown in Fig. 7) out to a radius of  $\sim 7$  pixels, slightly larger than the  $\sim 6$  pixels horizontally separating the centers of the microspectra. While we only fit the microspectra over a rectangle of 7-pixels wide, our model microspectrum extends out to a box roughly twice as wide. Subtracting all of the microspectra results in a small, negative residual as the broader boxes remove a few photons from the neighboring spectra. We iterate one time on the cube to remove cross talk. In practice, the effects of cross talk in CHARIS are extremely minor due to the spacing of our microspectra and our use of pinholes on the back of the lenslet array. The correction from an iteration to remove cross talk is generally  $\ll 1\%$ .

### 5.3 Residual Intensity and Read Noise Suppression

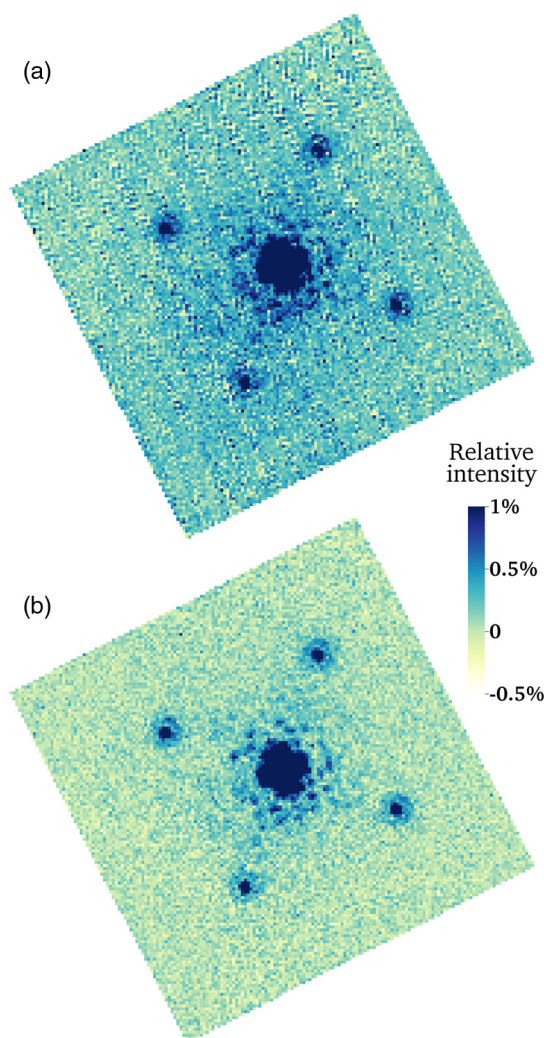
A particularly important feature of our  $\chi^2$  extraction is that it produces a 2-D model of the entire detector readout. Subtracting this model from the actual ramp produces a residual image that, in the limit of perfect models of the microspectra, is pure noise. Residuals from actual CHARIS ramps do show some systematics in the centers of the microspectra but are noise-dominated over much of the detector, both between microspectra and in the less-illuminated lenslets. CHARIS's H2RG



**Fig. 12** A single  $H$ -band slice through the same cube extracted three ways: (a)  $\chi^2$  extraction (Sec. 5.2), (b) optimal extraction (Sec. 5.1), and (c) an extraction using an unweighted 3-pixel-wide aperture. A  $\chi^2$  extraction intrinsically deconvolves the microspectra and the line-spread function, resulting in a more monochromatic image. Speckles in the optimal extraction slice are radially extended because of both the convolution with the line-spread function and interpolation onto a common wavelength array. The extraction using an unweighted aperture (c) is similar to optimal extraction but with slightly more noise and an inability to handle bad pixels.

detector has an excessive amount of read noise correlated among the readout channels, as discussed in Sec. 3.3 and shown in Fig. 5. We use our measurement of a residual intensity to model and remove the correlated component of the read noise, achieving a suppression approaching that shown in the lower-right panel of Fig. 5.

We fit for the correlated read noise as two patterns, one shared by the even readout channels and a second shared by the odd channels. We also fit for a scalar coupling between each channel and the appropriate noise pattern. We compute the correlated read noise using a trimmed mean over a user-specified fraction, by default 70%, of the pixels with the smallest fitted intensity relative to the read noise, i.e., the pixels where the residual is most dominated by the read noise. Each element of the correlated read noise pattern has 16 realizations on the detector (half of the 32 readout channels); this approach generally gives at least  $\sim 10$  pixels from which to calculate the pattern. We then scale the correlated read noise by our fitted couplings and subtract it from each readout channel.



**Fig. 13** A single  $H$ -band slice through a low-intensity cube extracted with our  $\chi^2$ -based algorithm (a) without and (b) with fitting out the correlated read noise and an undispersed background. The scales are the same on both images. Low-background regions are about 55% as noisy in the lower image as in the upper image. The noisier readout channels are visible as stripes in the slice without read noise suppression.

We find that our approach provides excellent suppression of the correlated component of the read noise while avoiding the addition of systematics back into the data. Figure 13 shows an example of a slice through an extracted cube with and without suppressing the read noise and fitting out an undispersed background. The image was taken through an ND filter to prevent saturation of the central star, resulting in low SNR over much of the image. The correlated read noise was severe on the date shown, but it is mostly removed by our algorithm.

We also enable the suppression of correlated read noise when using optimal extraction or aperture photometry. In this case, we use  $\chi^2$  extraction but save the read noise and (optionally) the undispersed background rather than the data cube. We then subtract the fitted read noise before performing the requested extraction algorithm. This approach has the same effectiveness at removing correlated read noise as simply using  $\chi^2$  extraction.

Our approach to read noise suppression works because of CHARIS's redundancy. While the detector has  $2048 \times 2048$  pixels, we extract only  $\sim 22$  spectral measurements for each of our  $\sim 135 \times 135$  illuminated lenslets, using  $\sim 10$  pixels on average to fit each spectral measurement. The problem is sufficiently overconstrained in that we can also fit two  $2048 \times 64$  noise patterns without facing significant degeneracies.

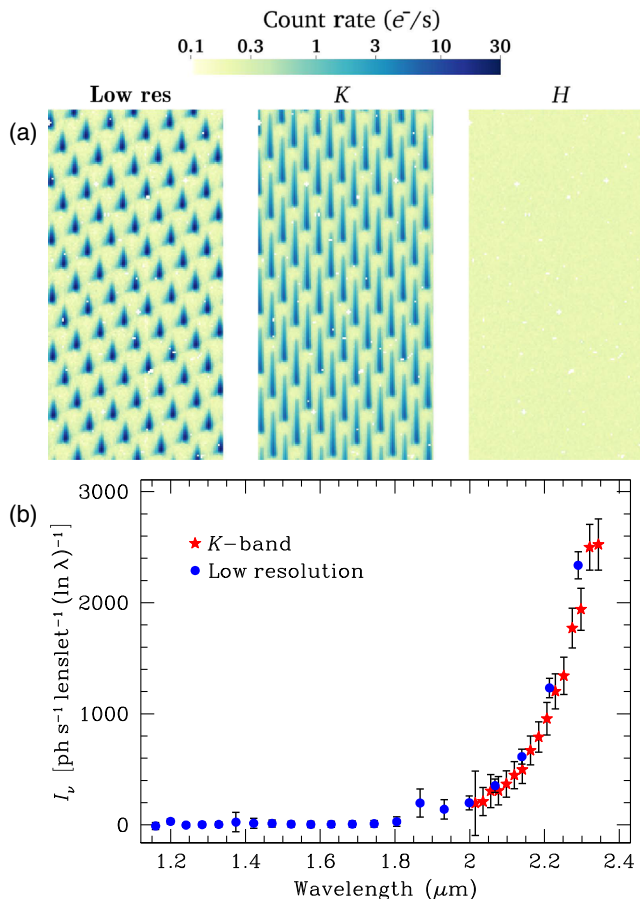
## 5.4 Background Subtraction

CHARIS ramps have a background overwhelmingly composed of light leaks and thermal photons in the  $K$ -band—the true dark current is negligible. The software does have the ability to subtract a background count rate from a 2-D ramp (i.e., a matched dark). This is common practice for near-infrared IFs, including OSIRIS<sup>7</sup> and GPI.<sup>22</sup> However, it is only worthwhile for CHARIS if a very high SNR dark frame is available in the same instrument configuration as the science data. Shot noise and read noise from the background frame will add to each image in an observing sequence, and, because the background has a single realization of the noise, it will add coherently to all images. The SNR in the background must be substantially higher than in each science frame (i.e., the integration time must be longer) to avoid this becoming a problem. Matching exposure times will result in the background subtraction contributing  $\sim$ half the noise to each science frame and most of the noise to a stack of science frames (because the same realization of noise is added to each frame).

The complexity of subtracting a background image is largely due to the fact that the background contains both a dispersed and an undispersed thermal component. The dispersed component is overwhelmingly due to longer wavelengths and is orders of magnitude lower with a filter in place that blocks  $K$ -band light. Microspectra from the thermal background are clearly visible on the detector in the low-resolution and  $K$ -band modes but are nearly invisible in the  $J$  and  $H$  bands or with the ND filter in place. Figure 14 shows these backgrounds over part of the detector in the low-resolution,  $K$ -, and  $H$ - bands. The dispersed component dominates the background in the low-resolution and  $K$ -bands, while the  $H$ -band shows a relatively uniform background of  $\sim 0.2 \text{ e}^- \text{ s}^{-1}$ . The microspectra in  $K$ -band are dispersed about four times as much as the low-resolution spectra, resulting in  $\sim 1/4$  of the peak intensity.

Any change in the alignment of the lenslet array will change the locations of the thermal microspectra on the array. The CHARIS extraction script can handle shifts in the location of the microspectra by applying a subpixel offset to the template





**Fig. 14** (a) Thermal backgrounds in the low-resolution, K-, and H-band modes. (b) Spectral intensity of the dispersed background in low-resolution and K-band modes showing its mean and standard deviation across the lenslet array. The background is composed of both a thermal component incident on the lenslet array and dispersed into microspectra by the prism and a nearly uniform, undispersed component. The dispersed component, which dominates the low-resolution and K-band backgrounds, is mostly from K-band light and is blocked by the J- and H-band filters.

PSFlets, but it cannot use this to shift the 2-D array of thermal microspectra. Any mismatch, even if it is just a fraction of a pixel, will degrade the ability of  $\chi^2$  extraction to fit the background-subtracted microspectra. Unfortunately, we lack the hardware (e.g., a mechanism to block the lenslet pinholes) to measure only the undispersed background.

It is easy to degrade data by subtracting a poor 2-D background. This is especially true of low-resolution data taken with the ND filter. Exposures with this filter are usually longer integrations to achieve good SNRs and, therefore, require long background images to perform a 2-D subtraction without adding noise. Subtracting a background taken with the broadband (rather than ND) filter results in a dispersed thermal background orders of magnitude too high and a negative inferred sky background in the K-band.

In the absence of a matched, high SNR and 2-D background, the software does have the ability to subtract an undispersed background lenslet-by-lenslet as part of a  $\chi^2$  extraction (Sec. 5.2) or combined with read noise suppression (Sec. 5.3). The dispersed background will still be included in the final data cube (though it can easily be removed by standard post-processing algorithms).

## 5.5 Extracted Cube

The script `extractcube` produces a flexible image transport system (FITS) file with the data cube (the intensity at each lenslet and at each wavelength) as the second Header/Data Unit (HDU 1) and the inverse variance of the data cube (the diagonal of the covariance matrix if using  $\chi^2$  extraction) as HDU 2. The first HDU, HDU 0, consists of only a header with key information about the observation and the data reduction. The last HDU, HDU 3, contains the original header of the file containing the raw reads.

The cube is defined on a logarithmic wavelength array. It is given by the nonstandard header keywords `LAM_MIN`, `LAM_MAX`, and `DLOGLAM`, with

$$\lambda_i = \text{LAMMIN} \times \exp[i \cdot \text{DLOGLAM}], \quad (8)$$

for  $i = 0, \dots, \text{NLAM}$ ; the spectral resolution of the extracted cube is  $R = 1/\text{DLOGLAM}$ . The wavelength scale is also given by the standard FITS keywords,<sup>56</sup> with `CTYPE3` = "AWAV-LOG". We use a logarithmic scale because of CHARIS's nearly wavelength-independent dispersion, and each pixel in the dispersion direction is a nearly constant increment in  $\log \lambda$ . A uniform spacing in wavelength would result in aliasing problems at the short wavelength end of the spectrum.

With  $\chi^2$  extraction, the data cube is never interpolated except over bad spectral measurements (which in any case have their inverse variance set to zero). With aperture photometry or optimal extraction, each lenslet's microspectrum has been interpolated to place it on a common wavelength array. In any of these cases the inverse variance does not capture the full spectral covariance. Still, it does capture lenslet-to-lenslet variations in the quality of the cube and prevents bad measurements from corrupting the images in postprocessing.

## 6 Software Parameters and Settings

To reduce CHARIS data, we provide a software package that accomplishes the two required tasks: wavelength calibration (through the script `buildcal`, see Sec. 4) and cube reduction (the script `extractcube`, see Sec. 5). The wavelength calibration only needs to point to a monochromatic flat taken as close as possible to the actual date that the science data were taken, ideally the same night. The user can choose whether or not to compute oversampled PSFlets, which are required to fit for subpixel shifts in the ramp as described in Sec. 5.2.

The cube extraction script, `extractcube`, has several parameters that the user can set; these will change the photometry and morphology of the reduced spatio-spectral datacube as described in Sec. 5 and Fig. 12. The set of parameters is presented in Table 2 and can be edited within a `.ini` text file that is read by the extraction software.

Many of the parameters in Table 2 are intended to be left as fixed. Indeed, the only parameter that a typical user will need to modify from its default value is `calibdir` (which must point to a local directory). Some should never be changed from their default values, while others may be modified to suit a particular reduction or a user's particular needs. Several of these parameters, however, allow the user to run a faster reduction, to change the treatment of the background, and to write intermediate data products.

Table 3 lists two sets of parameters, one intended for a quick reduction and the other for a more detailed reduction that removes read noise and performs a  $\chi^2$  extraction. The

Table 2 Extraction software parameters.

Parameter	Default	Allowed values	Description
Ramp parameters			
read_0	1	$[1, N_{\text{reads}} - 1]$	First read in the ramp to use.
read_f	None	$[\text{read\_0} + 1, N_{\text{reads}}]$ or None	Last read to use. If None, read_f will be the final read.
gain	2	$[0, \infty)$	Detector gain, $e^-/\text{count}$ , used to compute photon noise.
noisefac	0	$[0, 1)$	Additional fractional error in count rate to account for systematic errors in PSFlet models. Suggested values: 0.05 for $\chi^2$ extraction, 0 otherwise.
saveramp	False	True and False	Save the ramp as its own image?
Calibration parameters			
calibdir	—	String file path	Directory containing the files created by buildcal.
bgsub	False	True and False	Subtract a background/thermal ramp? If True, buildcal must have been called with background ramp(s). Should be False for ND frames.
mask	True	True and False	Mask bad pixels?
flatfield	True	True and False	Apply the pixel and lenslet flatfields?
fitshift	True	True and False	Fit for a subpixel shift of the microspectra? If True, buildcal must have computed oversampled PSFlets.
Extraction parameters			
R	30	$[1, \infty)$	Spectral resolution $R = (\delta \ln \lambda)^{-1} = \lambda/\delta\lambda$ of the extracted cube. Recommended: 30 for low res, 100 for high res.
method	lstsq	lstsq, optext, apphot3, apphot5	Method used to extract cube: $\chi^2$ extraction, optimal extraction, or aperture photometry across 3 or 5 pixels.
refine	True	True and False	Iterate once to remove cross talk? Only used if method is lstsq.
suppressrn	True	True and False	Use the method of Sec. 5.3 to remove read noise?
minpct	70	$(0, 100)$	Minimum percentage of pixels used to estimate read noise (only used if suppressrn is True).
fitbkgn	True	True and False	Fit and remove an undispersed background lenslet-by-lenslet?
smoothmask	True	True and False	Interpolate over bad lenslets to display a smooth cube? The inverse variance of the bad measurements remains zero.
saveresid	False	True and False	Save the 2-D residual from the ramp? Only used if method is lstsq.
maxcpus	None	$[1 - N_{\text{cpus}}, N_{\text{cpus}}]$ or None	Maximum number of threads to use. If None, use all threads. If negative, it is the number of threads to reserve.

parameters read\_0, read\_f, gain, mask, and flatfield are the same in both reductions; we generally recommend that the user never change these values. Changing either flatfield or mask to False, for example, will save a negligible amount of computational effort. To extract a cube with bgsub, a thermal background must have been computed by buildcal, while using fitshift requires oversampling the PSFlets in buildcal. The parameter smoothmask is intended only for cosmetic purposes; it replaces spectral measurements that are much noisier than their neighbors with

values taken from a smoothed cube. To avoid biasing the results, any modified intensities have their corresponding inverse variances set to zero. The parameters refine and minpct are ignored in the fast and rough reduction, refine because method is not lstsq and minpct because suppressrn is False. Table 3 indicates these extraneous parameters with ellipses.

One of the more problematic aspects of the reduction is background subtraction, as discussed in Sec. 5.4. Subtracting a good match to the 2-D thermal background (comprising both the microspectra and undispersed background) can



**Table 3** Suggested parameter settings.

Parameter	Fast and rough reduction	Slower and better reduction
read_0	1	1
read_f	None	None
gain	2	2
noisefac	0	0.05
bgsub	<sup>a</sup> False	<sup>a</sup> False
mask	True	True
flatfield	True	True
fitshift	False	True
R	<sup>b</sup> 30 or 100	<sup>b</sup> 30 or 100
method	optext	lstsq
refine	...	True
suppressrn	False	True
minpct	...	70
fitbkgnd	<sup>a</sup> False	<sup>a</sup> True
smoothermask	True	True

<sup>a</sup>See Secs. 5.4 and 6 for a discussion. Never set both bgsub and fitbkgnd to True.

<sup>b</sup>30 in low-resolution mode and 100 in  $J$ ,  $H$ , or  $K$ .

improve the reduction. However, it is easy to degrade the final cube by subtracting a noisy background or one that is a poor match to either the thermal microspectra or the undispersed background. For this reason, we generally recommend that a user set bgsub to False. If the user has a background image matching the observing mode and with at least a factor of a few longer integration time than any of the science frames, then setting bgsub to True may improve the reduction. If bgsub is set to False, then the undispersed background may be fit and removed by setting both fitbkgnd and suppressrn to True or by setting fitbkgnd to True with method equal to lstsq. The user should never attempt to remove the background twice by setting both bgsub and fitbkgnd equal to True.

## 7 Software Performance and Scaling

The CHARIS software is written in Python and Cython<sup>57</sup> and uses the extensive libraries available in the NumPy, SciPy,<sup>58</sup> and Astropy<sup>59</sup> packages. In addition to the parallelization performed natively in these packages, we have used the multiprocessing module to parallelize the construction of model PSFlets and OpenMP to parallelize many of the Cython routines.

We have tested the performance and scaling of the software on a compute server with two 12-core 2.5-GHz Intel Xeon processors. We use maxproc to set the maximum number of threads to allocate and test the two sets of parameters given in Table 3. We use a 41-read ramp (about a 60-s exposure);

a shorter exposure would require correspondingly less time to compute the ramp.

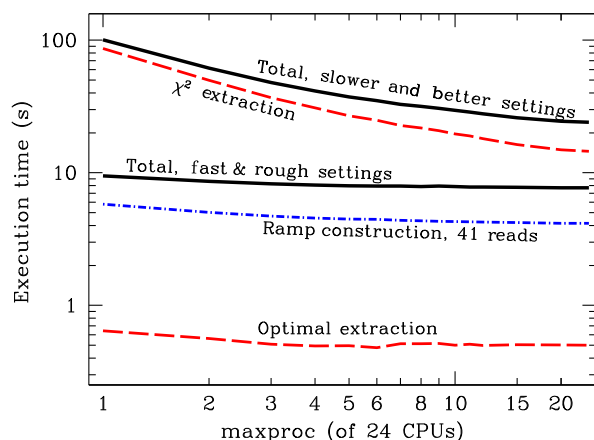
Figure 15 shows the results. The software has a few seconds of overhead from the initial module imports and from astropy computations of important parameters of the observation (such as the parallactic angle). The computation of the ramp requires a large amount of I/O and scales with the number of reads; using only the first 10 of the 41 reads reduces the computational cost by about a factor of 4. For the fast and rough reduction, OpenMP provides only a slight performance gain, and the cube extraction is a minor component of the total computational cost. The slower and better reduction, however, sees large performance gains from parallelizing the  $\chi^2$  extraction step, scaling reasonably well up to  $\sim 10$  CPUs. This step requires the cube to be extracted three times: an initial extraction, a second extraction once a first estimate of the correlated noise has been removed (Sec. 5.3), and a third after removing updated estimates of the correlated read noise and spectral cross talk. Setting refine and suppressrn to False (which we do not recommend) would reduce the computational cost of the  $\chi^2$  extraction step by nearly a factor of 3. Using optimal extraction but setting suppressrn to True would require at least two  $\chi^2$  extractions (three if also computing an undispersed background). In this case, the total computational time would be almost identical to the results shown here for the slower and better reduction with  $\chi^2$  extraction.

## 8 Discussion and Conclusions

In this paper, we have presented the data reduction pipeline for the CHARIS IFS. The software begins with the raw detector reads, corrects for an artifact in the first read, and then reconstructs the count rate at each pixel using either UTR weighting or the full nonlinear response. This approach handles saturation gracefully as long as there are at least a few reads before a pixel saturates, and it uses only those reads that are uncontaminated by bleeding from a neighboring saturated pixel. Fitting the full nonlinear response to pixels that exceed  $\sim 10\%$  of well capacity adds a negligible amount to the required runtime.

There are two steps after constructing the ramp: first, all of the calibration files must be built from a narrowband flatfield image, and, second, the data cube must be extracted. Our calibration step uses a range of data products that we have built from early lab data and from detailed image sequences taken with a supercontinuum source and a tunable filter. These products consist of the detector flatfield, the lenslet flatfield, a bad pixel mask, a reference wavelength solution (a wavelength-dependent mapping from lenslet to pixel coordinates), and reconstructed position- and wavelength-dependent lenslet PSFs. We supply a script buildcal with the software; buildcal takes a narrowband filter from the same night as the data to be reduced and computes a perturbation to the wavelength solution and resampled PSFlets for each wavelength and position to be extracted.

Once the calibration step has been completed, the script extractcube extracts the data cube. This script operates on a sequence of reads and requires a set of input parameters that the user may set using a .ini file. It first constructs the count rate at each pixel from the sequence of reads and computes basic parameters of the observation using keywords in the FITS header. Next, buildcal optionally computes a sub-pixel shift in the positions of the microspectra and finally extracts the data cube.



**Fig. 15** Performance and scaling results on a server with two 12-core 2.5-GHz Intel Xeon processors for the two suggested parameter sets in Table 3. The  $\chi^2$  extraction step scales reasonably well up to  $\sim 10$  cores and dominates the computational cost of our slower, better reduction. The fast and rough reduction is dominated by the construction of the ramp (which requires a lot of I/O and scales poorly), overheads, and astropy computations of observational parameters.

We implement three cube extraction algorithms: aperture photometry, optimal extraction, and  $\chi^2$  extraction. The GPI and SPHERE pipelines currently use aperture photometry as their primary extraction method.<sup>54,55,22</sup> While a least-squares inversion technique has been implemented for GPI,<sup>46</sup> it has not yet equaled the performance of simple aperture photometry. Project 1640’s pipeline<sup>49</sup> comes closest to our approach; however, after reconstructing the lenslet PSFs, it uses a weighted sum with the PSFlet templates rather than a  $\chi^2$  inversion. This decision increases the coupling among neighboring wavelengths, as it effectively performs a second convolution with the line-spread function.

All of the aperture photometry and optimal extraction techniques require interpolation to place the microspectra onto a common wavelength array. This interpolation couples neighboring wavelengths and degrades the data cube, while aperture photometry also requires bad data to be fixed. Our  $\chi^2$  extraction algorithm avoids all of these drawbacks and performs a deconvolution of the microspectra with the line-spread function, resulting in much more monochromatic slices through the data cube. The  $\chi^2$  algorithm produces a full 2-D residual map that allows us to estimate and remove correlated read noise, which results in a dramatic improvement to CHARIS data. This algorithm also allows us to fit out and remove spectral cross talk and an undispersed lenslet-by-lenslet background, and it produces a full covariance matrix for each lenslet’s microspectrum (though we presently save only the diagonal elements).

The final data product of the CHARIS pipeline is an extracted  $(x, y, \lambda)$  data cube with inverse variances for each spatial and spectral measurement. This cube must subsequently be calibrated, using satellite spots induced by SCEXAO<sup>26</sup> or some other method. The pipeline currently saves an estimated transformation from lenslet to sky coordinates; we defer a full astrometric calibration to a future paper. We caution against naïvely interpolating our data cube, as doing so would mix spatial and spectral elements that could have very different uncertainties and will inevitably discard information in the cube.

The CHARIS software is written in Python and Cython and is freely available on github: <https://github.com/PrincetonUniversity/charis-dep>.

University/charis-dep. It is parallelized, taking anywhere from a couple of seconds to run on a short ramp with a rough reduction to  $\sim 100$  s on a longer ramp with a detailed reduction using a single processor. With a moderately powerful computer server, the most detailed reductions may be completed in  $\sim 20$  to 30 s/frame. The CHARIS pipeline has a documentation page that we maintain at: <http://princetonuniversity.github.io/charis-dep/>. The data cubes that it produces are suitable inputs for high-contrast image processing, a topic that we will address in forthcoming work.

## Acknowledgments

This work was performed in part under contract with the Jet Propulsion Laboratory funded by NASA through the Sagan Fellowship Program executed by the NASA Exoplanet Science Institute. This work was performed in part under a Grant-in-Aid for Scientific Research on Innovative Areas from MEXT of the Japanese government (No. 23103002). This research was based on data collected at the Subaru Telescope, which is operated by the National Astronomical Observatories of Japan. The authors wish to recognize and acknowledge the very significant cultural role and reverence that the summit of Mauna Kea has always had within the indigenous Hawaiian community. We are most fortunate to have the opportunity to conduct observations from this mountain.

## References

1. C. Vanderriest, “A fiber-optics dissector for spectroscopy of nebularities around quasars and similar objects,” *Publ. Astron. Soc. Pac.* **92**(550), 858 (1980).
2. R. Bacon et al., “3D spectrography at high spatial resolution. I. Concept and realization of the integral field spectrograph TIGER,” *Astron. Astrophys. Suppl. Ser.* **113**, 347 (1995).
3. N. Drory et al., “The MaNGA integral field unit fiber feed system for the Sloan 2.5 m telescope,” *Astron. J.* **149**, 77 (2015).
4. R. Bacon et al., “The MUSE second-generation VLT instrument,” *Proc. SPIE* **7735**, 773508 (2010).
5. G. Bagnasco et al., “Overview of the near-infrared spectrograph (NIRSpec) instrument on-board the James Webb Space Telescope (JWST),” *Proc. SPIE* **6692**, 66920M (2007).
6. F. Eisenhauer et al., “SINFONI-Integral field spectroscopy at 50 milli-arcsecond resolution with the ESO VLT,” *Proc. SPIE* **4841**, 1548 (2003).
7. J. Larkin et al., “OSIRIS: a diffraction limited integral field spectrograph for Keck,” *Proc. SPIE* **6269**, 62691A (2006).
8. C. Marois et al., “GPI PSF subtraction with TLOCI: the next evolution in exoplanet/disk high-contrast imaging,” *Proc. SPIE* **9148**, 91480U (2014).
9. W. B. Sparks and H. C. Ford, “Imaging spectroscopy for extrasolar planet detection,” *Astrophys. J.* **578**(1), 543–564 (2002).
10. T. S. Barman et al., “Clouds and chemistry in the atmosphere of extrasolar planet HR8799b,” *Astrophys. J.* **733**, 65 (2011).
11. T. Currie et al., “Direct imaging and spectroscopy of a candidate companion below/near the deuterium-burning limit in the young binary star system, ROXs 42b,” *Astrophys. J. Lett.* **780**(2), L30 (2014).
12. S. Hinkley et al., “The  $\kappa$  Andromedae system: new constraints on the companion mass, system age, and further multiplicity,” *Astrophys. J.* **779**(2), 153 (2013).
13. Q. M. Konopacky et al., “Detection of carbon monoxide and water absorption lines in an exoplanet atmosphere,” *Science* **339**(6126), 1398–1401 (2013).
14. M. W. McElwain et al., “First high-contrast science with an integral field spectrograph: the substellar companion to GQ Lupi,” *Astrophys. J.* **1**, 505–514 (2007).
15. B. A. Macintosh et al., “The Gemini planet imager: from science to design to construction,” *Proc. SPIE* **7015**, 701518 (2008).

16. R. U. Claudi et al., “SPHERE IFS: the spectro differential imager of the VLT for exoplanets search,” *Proc. SPIE* **7014**, 70143E (2008).
17. S. Hinkley et al., “A new high contrast imaging program at Palomar observatory,” *Publ. Astron. Soc. Pac.* **123**(899), 74–86 (2011).
18. B. Macintosh et al., “Discovery and spectroscopy of the young jovian planet 51 Eri b with the Gemini Planet Imager,” *Science* **350**(6256), 64–67 (2015).
19. K. R. Stapelfeldt et al., “Exo-C: a probe-scale space mission to directly image and spectroscopically characterize exoplanetary systems using an internal coronagraph,” *Proc. SPIE* **9143**, 91432K (2014).
20. M. W. McElwain et al., “PISCES: an integral field spectrograph technology demonstration for the WFIRST coronagraph,” *Proc. SPIE* **9904**, 99041A (2016).
21. D. Spergel et al., Wide-field infrared survey telescope-astrophysics focused telescope assets WFIRST-AFTA 2015 report, (2015).
22. M. D. Perrin et al., “Gemini planet imager observational calibrations XI: pipeline improvements and enhanced calibrations after two years on sky,” *Proc. SPIE* **9908**, 990837 (2016).
23. M. D. Perrin et al., “Gemini planet imager observational calibrations I: overview of the GPI data reduction pipeline,” *Proc. SPIE* **9147**, 91473J (2014).
24. Y. Minowa et al., “Performance of Subaru adaptive optics system AO188,” *Proc. SPIE* **7736**, 77363N (2010).
25. N. Jovanovic et al., “The Subaru coronagraphic extreme adaptive optics system: enabling high-contrast imaging on solar-system scales,” *Publ. Astron. Soc. Pac.* **127**(955), 890–910 (2015).
26. N. Jovanovic et al., “Artificial incoherent speckles enable precision astrometry and photometry in high-contrast imaging,” *Astrophys. J. Lett.* **813**(2), L24 (2015).
27. M. W. McElwain et al., “Scientific design of a high contrast integral field spectrograph for the Subaru telescope,” *Proc. SPIE* **8446**, 84469C (2012).
28. M. A. Peters et al., “Conceptual design of the coronagraphic high angular resolution imaging spectrograph (CHARIS) for the Subaru telescope,” *Proc. SPIE* **8446**, 84467U (2012).
29. M. B. Galvin et al., “The mechanical design of CHARIS: an exoplanet IFS for the Subaru telescope,” *Proc. SPIE* **9151**, 91513G (2014).
30. M. A. Peters-Limbach et al., “The optical design of CHARIS: an exoplanet IFS for the Subaru telescope,” *Proc. SPIE* **8864**, 88641N (2013).
31. T. D. Groff et al., “Laboratory testing and performance verification of the CHARIS integral field spectrograph,” *Proc. SPIE* **9908**, 99080O (2016).
32. R. M. Smith et al., “A theory for image persistence in HgCdTe photodiode,” *Proc. SPIE* **7021**, 70210J (2008).
33. S. H. Moseley et al., “Reducing the read noise of H2RG detector arrays: eliminating correlated noise with efficient use of reference signals,” *Proc. SPIE* **7742**, 77421B (2010).
34. A. M. Fowler and I. Gatley, “Demonstration of an algorithm for read-noise reduction in infrared arrays,” *Astrophys. J.* **353**, L33 (1990).
35. L. Dressel, *Wide Field Camera 3 Instrument Handbook v. 9.0*, Space Telescope Science Institute, Baltimore, Maryland (2017).
36. G. Finger et al., “Performance evaluation, readout modes, and calibration techniques of HgCdTe Hawaii-2RG mosaic arrays,” *Proc. SPIE* **7021**, 70210P (2008).
37. B. J. Rauscher et al., “Detectors for the James Webb space telescope near-infrared spectrograph. I. Readout mode, noise model, and calibration considerations,” *Publ. Astron. Soc. Pac.* **119**(857), 768–786 (2007).
38. D. J. Fixsen et al., “Cosmic-ray rejection and readout efficiency for large-area arrays,” *Publ. Astron. Soc. Pac.* **112**, 1350–1359 (2000).
39. M. Robberto, “A generalized least square algorithm to process infrared data taken in non-destructive readout mode,” *Proc. SPIE* **9143**, 91433Z (2014).
40. J. D. Offenberger et al., “Validation of up-the-ramp sampling with cosmic-ray rejection on infrared detectors,” *Publ. Astron. Soc. Pac.* **113**(780), 240–254 (2001).
41. C. M. Bacon et al., “Further characterization of Rockwell Scientific LWIR HgCdTe detector arrays,” *Proc. SPIE* **5563**, 35 (2004).
42. T. D. Brandt et al., “New techniques for high-contrast imaging with ADI: The ACORNS-ADI SEEDS data reduction pipeline,” *Astrophys. J.* **764**, 183 (2013).
43. B. Kubik et al., “Impact of noise covariance and nonlinearities in NIR H2RG detectors,” *Proc. SPIE* **9154**, 91541B (2014).
44. J. Anderson and I. R. King, “Toward high-precision astrometry with WFC2. I. Deriving an accurate point-spread function,” *Publ. Astron. Soc. Pac.* **112**(776), 1360–1382 (2000).
45. P. Ingraham et al., “Gemini planet imager observational calibrations III: empirical measurement methods and applications of high-resolution microlens PSFs,” *Proc. SPIE* **9147**, 91477K (2014).
46. Z. H. Draper et al., “Gemini planet imager observational calibrations IX: least-squares inversion flux extraction,” *Proc. SPIE* **9147**, 91474Z (2014).
47. L. B. Lucy, “An iterative technique for the rectification of observed distributions,” *Astron. J.* **79**, 745 (1974).
48. S. G. Wolff et al., “Gemini planet imager observational calibrations IV: wavelength calibration and flexure correction for the integral field spectrograph,” *Proc. SPIE* **9147**, 91477H (2014).
49. N. Zimmerman et al., “A data-cube extraction pipeline for a coronagraphic integral field spectrograph,” *Publ. Astron. Soc. Pac.* **123**, 746–763 (2011).
50. K. Horne, “An optimal extraction algorithm for CCD spectroscopy,” *Publ. Astron. Soc. Pac.* **98**(604), 609 (1986).
51. A. Baranne et al., “ELODIE: a spectrograph for accurate radial velocity measurements,” *Astron. Astrophys. Suppl. Ser.* **119**, 373–390 (1996).
52. A. S. Bolton et al., “Spectral classification and redshift measurement for the SDSS-III baryon oscillation spectroscopic survey,” *Astron. J.* **144**, 144 (2012).
53. M. C. Cushing, W. D. Vacca, and J. T. Rayner, “Spectool: a spectral extraction package for SpeX, a 0.8–5.5 micron cross-dispersed spectrograph,” *Publ. Astron. Soc. Pac.* **116**, 362–376 (2004).
54. D. Mesa et al., “Performance of the VLT Planet Finder SPHERE-II. Data analysis and results for IFS in laboratory,” *Astron. Astrophys.* **576**, A121 (2015).
55. A. Pavlov et al., “SPHERE data reduction and handling system: overview, project status, and development,” *Proc. SPIE* **7019**, 701939 (2008).
56. E. W. Greisen et al., “Representations of spectral coordinates in FITS,” *Astron. Astrophys.* **446**(2), 747–771 (2006).
57. L. Dalcin et al., “Cython: the best of both worlds,” *Comput. Sci. Eng.* **13**, 31 (2011).
58. S. Van Der Walt, S. C. Colbert, and G. Varoquaux, “The NumPy array: a structure for efficient numerical computation,” *Comput. Sci. Eng.* **13**(2), 22–30 (2011).
59. T. P. Robitaille et al., “Astropy: a community python package for astronomy,” *Astron. Astrophys.* **558**, A33 (2013).

Biographies for the authors are not available.